USERS

phpAdsNew GESTIONANDO LA PUBLICIDAD DE NUESTRO SITIO

AR $6,90
Recargo envío al interior $0,20
MX $22

LINUX 06

comunidad libre

OPTIMIZACION

MEDIR EL RENDIMIENTO Y OPTIMIZAR APLICACIONES

LAS MEJORES DISTROS PARA HACER TUNNING

ADAPTAR EL KERNEL A NUESTRO HARD

CONFIGURAR SERVICIOS

ADEMAS

LAS ULTIMAS NOTICIAS SOBRE EL MUNDO GNU/LINUX
TRUCOS ★ CONSULTAS
Y MUCHO MAS!!!!

Soluciones

★ APLICANDO PARCHES AL KERNEL 2.6
★ CONFIGURAR PERMISOS CON CHMOD ★ FREEBSD, UNA ALTERNATIVA A GNU

Actualidad

★ FIRESTARTER 1.0, UN SOFT PARA EL MANEJO DE REGLAS DE FIREWALL
★ X.ORG, EL REEMPLAZO DE XFREE86 ★ REPRODUCTORES MULTIMEDIA

Expertos

★ COURIER-MTA, EL SERVIDOR DE CORREO ELECTRONICO
★ COMO PROGRAMAR UN SERVIDOR DE CHAT EN C

Nessus Security Scanner

COMO AUDITAR UN SERVIDOR E IMPLEMENTAR MEDIDAS NECESARIAS PARA CORREGIR SUS VULNERABILIDADES

ISBN 987-526-260-9

LINUX 06 comunidad libre

nuestros expertos

Ariel Corgatelli ★ arielcorg@tectimes.com
Pasa horas y horas probando programas y convenciendo a la gente de que las alternativas libres son tan buenas o mejores que las aplicaciones populares propietarias.

Demian Pablo Alonso ★ demon@tectimes.com
Programador profesional, Demian es el encargado de enseñarnos las mejores técnicas de codificación y de mostrarnos las más avanzadas herramientas para el desarrollo de aplicaciones.

Arturo Busleiman ★ buanzo@buanzo.com.ar
Un usuario de GNU/Linux y software libre de la vieja escuela. "El buanzo" (como se lo conoce en el ámbito) comparte todos sus conocimientos de administración en la sección redes.

Martín Ramos Monso ★ mramos@infocomercial.com
Autor de un libro de programación en lenguaje PHP (publicado por esta editorial), Martín es un experto en desarrollo de sitios web utilizando herramientas de código abierto.

Agustín Casiva ★ casivaagustini@gmail.com
Un nuevo colaborador de GNU/Linux USERS. Agustín se encargará de contarnos toda su experiencia en el uso de sistemas de bases de datos y aplicaciones para pymes.

Guido Lorenzutti ★ glorenzutti@arnet.com.ar
Otro usuario de GNU/Linux de la vieja escuela. Guido es un administrador de sistemas UNIX experto en seguridad, y compartirá todos sus conocimientos sobre cómo proteger nuestros servidores.

Juan P. Firrincieli ★ quasarfreak@gmail.com
Entusiasta usuario de GNU/Linux en todo lo que sea aplicaciones visuales. Juan Pablo nos mostrará los mejores juegos y las aplicaciones más útiles para el hogar y la oficina.

Juan Rodríguez ★ jrodriguez@linux.org.ar
Fanático de GNU/Linux y administrador del sitio web del LugAr (Linux User Group Argentina). Juan es un experto administrador de sistemas.

Jorge Rodríguez ★ admin@salux.org
"Tio JR" (como le gusta que lo llamen) es un médico apasionado de GNU/Linux y el mundo del software libre. Es uno de los encargados del grupo Biolinux.

Los lectores que no estén suscriptos a la revista podrán de todas formas bajar los programas del CD desde **linux.tectimes.com**.

editorial

Mucha gente se queja de que cuando instala una nueva versión de GNU/Linux en su computadora, el sistema no funciona a la misma velocidad que su clásico Windows 98 (o la versión que quieran, la 98 sigue siendo la más popular). La cuestión es, en parte, cierta, ya que muchas distribuciones de propósito general (como Mandrake, Fedora o SuSE) están desarrolladas siguiendo los estándares de hardware actuales, y muy poca gente se mantiene siempre actualizada en este aspecto. ¿Quiere esto decir que debemos resignarnos a la lentitud? ¡De ninguna manera! Una de las particularidades del sistema operativo GNU/Linux es la flexibilidad: nosotros podemos "modificar" el funcionamiento de una distribución para que se adecue a nuestras necesidades. De esta forma, podemos hacer que un Mandrake 10.1 que requiere 128 MB de RAM como mínimo para funcionar, lo haga sin problemas con 64 MB, simplemente haciendo algunos toquecitos por aquí y otros por allá. Ese es el arte del "tuning", de la optimización. En el mundo de GNU/Linux, es muy común tener la impresión de que siempre es posible afinar un poquito más nuestro sistema para que funcione mejor en nuestro hardware. Por esta razón, armamos una nota de tapa dedicada exclusivamente a este "arte". Les pedimos consejos a los diferentes colaboradores de la revista y se los dejamos servidos en bandeja para que puedan comenzar a afinar sus sistemas. Pero cuidado, presentamos sólo los consejos que consideramos más importantes, aunque hay muchísimo por hacer. Si descubren algo realmente interesante, los invito a que nos lo cuenten para que el resto de los lectores pueda conocer el truco y aprovecharlo. Espero que disfruten de esta nota y de toda la revista. Hasta la próxima.

★ **Héctor Facundo Arena**

actualidad

soluciones

expertos

staff

Coordinador Editorial ★ Miguel Lederkremer
Asesor Editorial ★ Fernando Casale
Asesor de Diseño ★ Flavio Burstein
Diagramación ★ Carlos Peralta ★ Salvador Curutchet
Proveedor de Diseño ★ Juan Pablo Gurzi
Corrección ★ Magdalena Porro
Asesor Editorial General ★ Gabriel Pleszowski
Asesor de Diseño ★ Frank Sozzani
Asesor de Marketing ★ Benito de Miguel

Todas las marcas mencionadas son propiedad de sus respectivos dueños. Impreso en Kollor Press S.A. Copyright © MMIV MP Ediciones S.A., Moreno 2062, C1094ABF, Ciudad de Buenos Aires, Argentina. Tel.: (54-11) 4959-5000. Fax: (54-11) 4954-1791. E-mail: correo@tectimes.com. Hecho el depósito que marca la ley. Esta publicación no puede ser reproducida, ni en todo ni en parte, ni registrada en o transmitida por un sistema de recuperación de información, en ninguna forma ni por ningún medio, sea mecánico, fotoquímico, electrónico, magnético, electroóptico, por fotocopia o cualquier otro, sin el permiso previo y por escrito de esta casa editorial.

16 OPTIMIZACION: COMO MEDIR EL RENDIMIENTO, LAS MEJORES DISTROS PARA HACER TUNNING, ADAPTAR EL KERNEL A NUESTRO HARDWARE, CONFIGURAR SERVICIOS Y MUCHO MAS.

10 Firestarter
Evaluamos un magnífico software gráfico para el manejo de reglas de firewall, ideal para aquellos linuxeros que no tengan ganas de complicarse con puertos, hosts, y demás.

12 X.org
Un análisis a fondo del reemplazo de Xfree86. Se trata de una implementación del sistema XWindow en código abierto y de distribución libre.

06 News
Una sección para que estén informados de todas las novedades del mundo GNU/Linux.

14 Reproductores multimedia
Las opciones para escuchar música y ver películas con GNU/Linux son cada vez más variadas, y es por ello que decidimos probarlas todas.

26 En el CD
Una colección de software seleccionado por nuestros expertos.

28 Trucos
Las mejores técnicas para el shell.

32 Consultas
En esta sección respondemos las preguntas que todos los meses nos envían los lectores a nuestro correo electrónico.

34 Shell: el comando chmod
Cómo dominar el sistema de permisos de archivos de GNU/Linux.

36 Shell: aplicando parches
Cómo actualizar el kernel de nuestro sistema operativo.

40 Sistemas operativos
FreeBSD, una alternativa a GNU.

52 Programación
Servidor de chat
En esta oportunidad veremos un ejemplo práctico de programación en C: desarrollaremos un pequeño servidor de chat. Este deberá permitir la conexión de cualquier cantidad de clientes, vía telnet, que se comunicarán entre ellos en una única sala virtual.

44 Desarrollo web
Todo sobre phpAdsNew.

48 Seguridad
Auditar un servidor con Nessus Security Scanner.

60 Servers
Courier-MTA: segunda parte.

64 Convivencia en paz
Un momento de reflexión acerca de la actualidad del mundo del software libre.

Publicidad: (54-11) 4959-5000; fax: (54-11) 4954-1791; publicidad@tectimes.com
Distribuidores: Argentina (Capital): Vaccaro Sánchez y Cía. C.S., Moreno 794 piso 9 (1091), Ciudad de Buenos Aires | **Chile:** Distribuidora Vía Directa S.A., Riquelme Nº 840, Santiago de Chile, tel. 688-7383 | **México:** CITEM S.A. de C.V., Av. del Cristo No. 101 Col. Xocoyahualco Tlalnepantla, Estado de México | **Bolivia:** Agencia Moderna Ltda., General Acha E-0132, C. de correo 462, Cochabamba, tel. 005914-422-1414 | **Uruguay:** Espert SRL, Ciudadela 1416, Montevideo
Venezuela: Distribuidora Continental Bloque de Armas, Edificio Bloque de Armas, piso 9°, Av. San Martín, cruce con final Av. La Paz, Caracas. | **Argentina Interior** - Perfil S.A. - Chacabuco 271 - Bs. As. | **Paraguay** - Selecciones S.A.C. - Coronel Gracia 225 - Asunción | **Perú** - Distribuidora Bolivariana - Av. República de Panamá 3635 - San Isidro, Lima

xxx / Coordinado por: Miguel Lederkremer - 1a ed. - Buenos Aires.
MP Ediciones, 2005. 64 p.; 28x20 cm. - (Linux Users, 6) **ISBN 987-526-260-9**
1. Hardware I. Lederkremer, Miguel, coor. CDD 004

LOS LECTORES SE HACEN LEER

Cartas

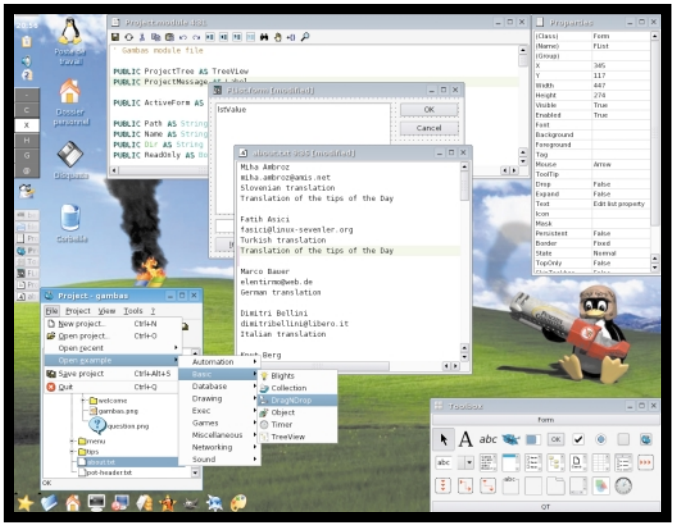
OPINIONES, COMENTARIOS, CRITICAS, SUGERENCIAS... EN FIN, TODO LO QUE USTEDES NOS ENVIAN POR MAIL, RESUMIDO EN UNA PAGINA.

PROGRAMACION VISUAL

De ★ Gustavo A. Karol
Mail ★ gak@sion.com
Asunto ★ ¡Visual Basic para Linux!

Saludos a todos. En USERS Linux #04, en la página 42, un lector pregunta por un Visual Basic para GNU/Linux. Hace un tiempo hallé algo que, al parecer, es similar. Lamentablemente, por falta de tiempo no pude probarlo, y quizá pasen 1.000 años hasta que tenga un rato libre. Se llama Phoenix, y si la memoria no me engaña, en Windows se denomina RapidQ. ¿Cómo lo conseguí? Buscando por palabras clave. Tal vez esta dirección les sea útil: **linuxshop.ru/linuxbegin/win-lin-soft-spanish/#7**. Espero que alguien pueda probarlo en 9.999 años o menos y envíe su experiencia a la revista para compartirla.

GNU/Linux USERS: Gustavo, ¡muchísimas gracias por la información! Les contamos a los lectores que encontramos otro proyecto similar a Visual Basic para GNU/Linux. Se llama **Gambas** (**gambas.sourceforge.net**) y está en una etapa muy avanzada de desarrollo. Ya es posible hacer programas realmente útiles con él. Para la próxima edición, tenemos preparada una nota sobre este tema y, quién les dice, más adelante quizás armemos algo sobre Phoenix también.



GAMBAS: NO SE VE IGUAL A VISUAL BASIC, PERO OFRECE MUCHAS HERRAMIENTAS DE DESARROLLO SIMILARES.

EXPERIENCIA CON WINMODEMS

De ★ Sebastián
Mail ★ seba771@hotmail.com
Asunto ★ Gracias por a nota de Winmódems

Les escribo para agradecerles por la nota que publicaron sobre winmódems. Yo fui quien la pedí, y ustedes acudieron en mi ayuda y en la ayuda de muchos otros usuarios. Como solucioné mi problema antes de recibir la revista, creo conveniente contarles mi experiencia, que puede ser útil para otros usuarios. Tenía un módem con chipset Agere (Lucent) SV92P-T00, y la herramienta Scanmodem me decía que, hasta el momento, no era soportado. Me cansé de pelear y no logré hacerlo andar.
Por eso decidí comprar otro módem, y averigüé que los de Intel eran muy requeridos. Me puse a indagar un poco y ¡oh sorpresa!, algunos chipsets de Intel tienen los drivers para Linux en la misma página de la compañía. Hay tres chipsets de Intel con soporte para Linux: 537EA, 536EP y 537EP. Comencé a revisar en MercadoLibre y en De Remate para ver qué conseguía, y encontré un Encore con chipset 537EP. Lo compré, bajé los drivers de Intel, descomprimí, instalé y voilá. ¡¡Al fin!!
Hay muchas empresas que actualmente tienen este chipset: Encore, Smart Link, etc. La distribución que uso es RedHat 9.0 y estoy por migrar a Fedora Core 3.
El chip del módem dice: FA82537EP, y el link de donde podrán bajar el driver (no compilado) es
http://downloadfinder.intel.com/scripts-df/Detail_Desc.asp?agr=Y&ProductID=1230&DwnldID=680.
Aunque solucioné mi inconveniente por mis propios medios, les agradezco mucho que sigan teniendo en cuenta a los usuarios que todavía nos movemos con dial up y no queremos gastar mucho dinero comprando un módem externo o un fullmódem.

GNU/Linux USERS: ¡Muchas gracias por tus comentarios! Usuarios con vocación de colaboración como vos son los que llevaron al Software Libre a su exitosa situación actual y los que lo llevarán más lejos todavía. Y sí, por supuesto que seguiremos teniendo en cuenta a usuarios de estas tecnologías tan "raras".

¡QUEREMOS SUS OPINIONES!

Recuerden que pueden enviarnos sus comentarios, opiniones y sugerencias a nuestra dirección de correo electrónico (linuxusers@tectimes.com). Y no olviden que las páginas de esta revista están abiertas a quien quiera colaborar enviando sus notas, noticias o lo que sea.

EL HERALDO DE GNU/LINUX Y SOFTWARE LIBRE

"La Libertad"

NOTICIAS, NOVEDADES, LANZAMIENTOS DE NUEVAS VERSIONES... EN FIN, TODO LO QUE SUCEDIO EN EL ULTIMO MES, RESUMIDO EN DOS PAGINAS.

SE VIENE SLACKWARE 10.1

En el número pasado anunciábamos que Patrick Volckerding (el creador y actual encargado de la distribución Slackware) estaba atravesando serios problemas de salud. Por suerte, el mismo Patrick anunció hace unos días que se está sintiendo mucho mejor y que está retornando a su trabajo diario de mantener una de las mejores distribuciones de GNU/Linux de todos los tiempos. Según sus propias palabras, la versión 10.1 está a la vuelta de la esquina (se está poniendo al tanto de todo el trabajo retrasado) y ya está investigando sobre las modificaciones necesarias del kernel para la próxima versión 11 de Slackware. ¡Nos alegramos mucho, Patrick!

2004, UN GRAN AÑO PARA GNU/LINUX

Ya en los comienzos del 2005, nos ponemos a pensar cómo fue el 2004, y vemos que la experiencia ha sido realmente asombrosa. Quienes trabajamos a diario con Software Libre hemos visto muchas más empresas y particulares interesados en migrar a GNU/Linux o a algún otro proyecto de este tipo. Por si no nos creen, les contamos que la popular consultora IDC anunció que, según sus cálculos, el crecimiento de GNU/Linux en el ámbito de los servidores ha sido de un 34%. De ahora en más, se prevé una tasa anual de crecimiento del orden del 40% y, para el 2008, se espera que el mercado mueva alrededor de 35.000 millones de dólares. ¡Qué buena forma de empezar el año!

FESTIVAL DE INSTALACION DE SOFTWARE LIBRE LATINOAMERICANO

Se llevará a cabo el sábado 2 de abril de 2005, el propósito de éste evento es promover el uso de Software Libre y la integración de comunidades de usuarios de este tipo de soft en todos los países de Latinoamérica. Para lograrlo, se realizarán eventos simultáneos en diferentes ciudades, en los que los expertos de cada una instalará, de manera gratuita y totalmente legal, software libre en las computadoras que para ese efecto traigan todas las personas interesadas. El Festival está siendo organizado por BSD Venezuela, COLIBRÍ, TALUG y VELUG, que convocan a participar a otras comunidades de usuarios de Software Libre de la región. En este momento se ha confirmado la participación de 22 ciudades, nueve en Colombia, ocho en Venezuela y cinco en la Argentina, y es probable que otras más se agreguen próximamente. Dentro del software que se instalará se incluyen distribuciones de GNU/Linux y BSD, así como Software Libre para otras plataformas como Windows y MacOS. Para obtener más información, pueden visitar la página oficial del evento, ubicada en **ingenieria.ean.edu.co/~azul/svnwiki.cgi/colibri/fisl/default**.

LOS PAISES DE LATINOAMERICA SE UNEN PARA DIFUNDIR EL SOFTWARE LIBRE. ¡QUE BUENA IDEA!

¡TUX RACER EN LOS VIDEOJUEGOS!

Sin lugar a dudas, Tux Racer es uno de los juegos más populares en el mundo de GNU/Linux. Para los que no lo conocen, es un título en 3D en el cual controlamos a un pingüino que se desliza por las rápidas caídas de una colina nevada, corriendo carreras con otros pingüinos. En el camino, se encuentra con obstáculos y con otros de sus parientes que tratarán de impedir que llegue a su meta. Bueno, tal fue el éxito de este producto, que una empresa norteamericana decidió construir una máquina recreativa de salón con él. ¿Quieren comprarla para ponerla en algún rincón de su casa? ¡No hay problema! Con tan sólo U$$ 3.995 es de ustedes.

TUX RACER, EN PLENA CARRERA. SI NO LO JUGARON NUNCA, ¡NO SABEN LO QUE SE PIERDEN!

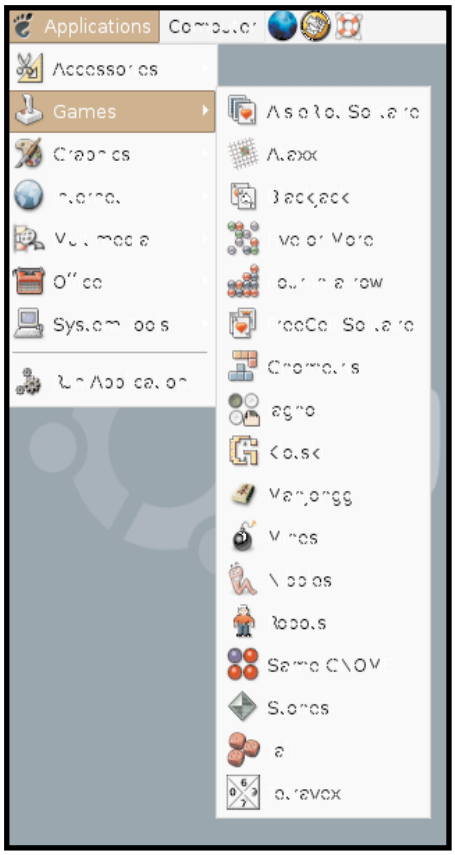
LINUX EN UNA GAMECUBE

Hace algunas ediciones, en esta misma sección, les mostrábamos una versión de UNIX corriendo en una Game Boy Advance. Bueno, eso no era suficiente y ¡vamos por más! Y sí, seguimos poniendo el sistema operativo del pingüino en todos lados. Esta vez, navegando por la Red nos encontramos con la página de este proyecto que, entre otras cosas, ya incluye librerías SDL que funcionan a la perfección. Más información, en **www.gc-linux.org**.

AQUI VEMOS EL SISTEMA CORRIENDO EN LA GAMECUBE. ¡INCREIBLE!

DEBIAN 4... ¿FALTA MUCHO? ¿POCO?

Como siempre, los rumores que se corrían durante el 2004 con respecto a la inminente salida de la versión 4 de Debian fueron sólo una cortina de humo. Nada pasó. De todas formas, en las mailing lists de Debian se ve que hay mucha actividad (y apuro) por tener la nueva versión lo antes posible. Para los impacientes, les contamos que la espera realmente valdrá la pena. En un hecho histórico sin precedentes para el proyecto, la nueva versión de Debian incluirá la última versión disponible de GNOME (la 2.8). ¡Vamos, muchachos, que el pueblo tiene hambre!



LA ULTIMA VERSION DE GNOME (LA 2.8) VENDRA INCLUIDA EN LA PROXIMA VERSION DE DEBIAN. PERO... ¿PARA CUANDO?

EL MEJOR SOFTWARE GRAFICO PARA EL MANEJO DE REGLAS DE FIREWALL

Firestarter 1.0

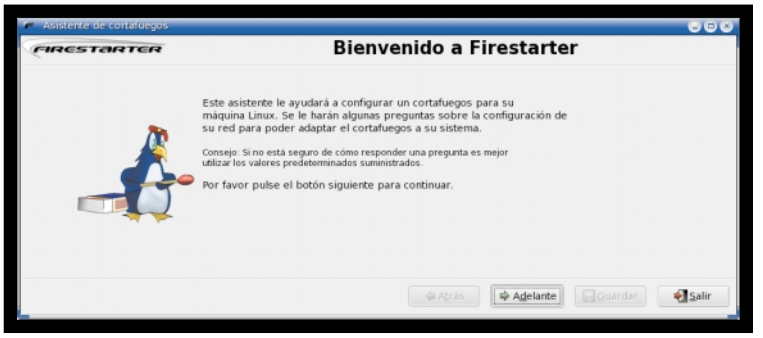
MUCHAS VECES ME PLANTEARON LA CONSULTA: ¿COMO PROTEJO MI SISTEMA? BUENO, HAY MUCHAS FORMAS DE HACERLO, Y UNA DE ELLAS ES PONER UN "MURO" O, MEJOR DICHO, UN FIREWALL, COMO SE LO CONOCE EN EL MEDIO INFORMATICO. ¿CREEN QUE ES DIFICIL DE CONFIGURAR? CON FIRESTARTER NO LO ES.

La necesidad de dar seguridad a un sistema informático que corre con GNU/Linux es por demás importante. Si bien hay muchas formas de proteger los datos y el sistema, siempre es importante controlar o, mejor dicho, limitar de algún modo el acceso a nuestro equipo tanto desde Internet como desde nuestra red interna. Esto puede lograrse muy fácilmente utilizando un firewall. La configuración de uno en GNU/Linux puede ser tan compleja como nosotros queramos y hay, en principio, dos maneras de realizarla: ingresando reglas en forma manual (desde el entorno de texto) o haciéndolo en forma automática. Firestarter nos permite generar reglas automáticamente si no sabemos mucho sobre manejo de Iptables. Además, cuenta con opciones de registro (logging) en lo que se refiere a conexiones a nuestro sistema. El firewall registra accesos, como así también intentos de violación, y todo puede traducirse en un archivo para examinarlo sin problemas.

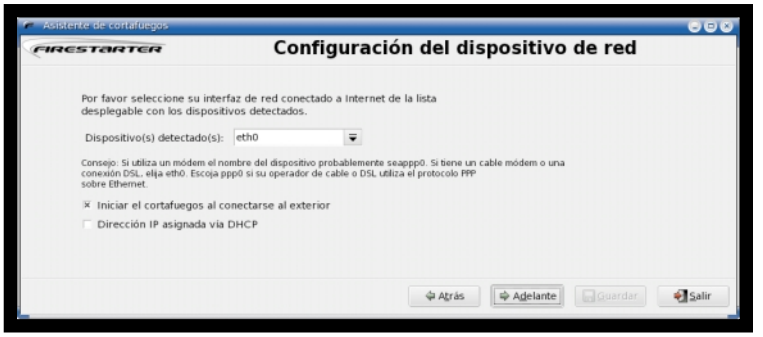
UNA SOLUCION PARA TODO EL MUNDO

Este soft tiene la capacidad de manejar perfectamente iptables sin necesidad de conocer host, IPs, puertos, etc. Con sólo hacer unos cuantos clics, nos ayuda a configurar toda la seguridad de nuestro sistema, en no más de cuatro pantallas.

La última versión es la 1.0 (al momento de redactar esta nota), y puede descargase desde **www.fs-security.com**. Allí encontraremos archivos en los formatos clásicos RPM y Tar.Gz para nuestra distribución. Si descargamos el paquete RPM, la instalación es muy simple y puede hacerse desde una terminal en texto de la siguiente manera:



AQUI PODEMOS VER LA PANTALLA DE BIENVENIDA AL ASISTENTE DE CONFIGURACION.



EL ASISTENTE NOS MOSTRARA LAS PLACAS DE RED DETECTADAS EN NUESTRO SISTEMA (ADEMAS DE UNA BREVE RESEÑA DE CADA UNA), PARA REALIZAR UNA CORRECTA CONFIGURACIÓN.

```
# rpm -i firestarter-1.0.0-1.rpm
```

También puede realizarse desde el entorno gráfico con el manejador de paquetes de turno.

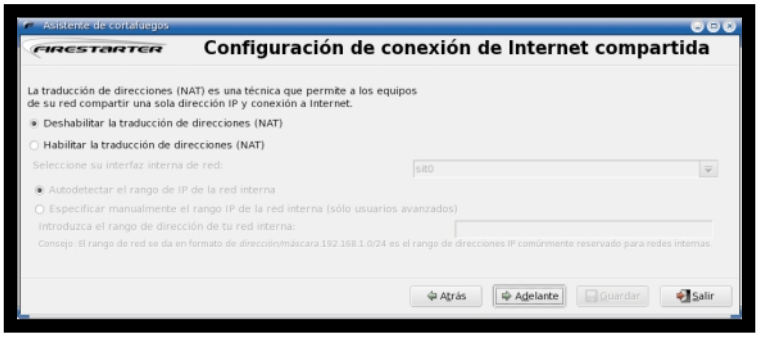
CONFIGURACION DEL SISTEMA

Este proceso se realiza de forma muy simple: con sólo llamar al comando **firestarter**, se presentará en pantalla un asistente de configuración con los siguientes pasos:

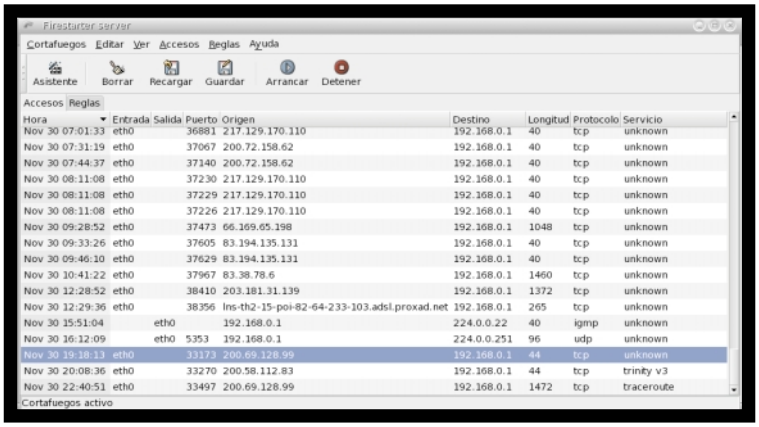
1) Como todo software que requiere modificar archivos de configuración en Linux, deberemos tener el password del administrador (root) para continuar.

2) El asistente nos da la bienvenida en español.

3) A continuación nos consulta cuál es nuestro dispositivo de red conectado a Internet; es posible que lo detecte correctamente.



¡MAGIA! CONTAMOS CON LA POSIBILIDAD DE ACTIVAR EL PROXY VIA NAT, CON SOLO UNOS POCOS CLICS.



PODEMOS VER UN INFORME DE ACTIVIDADES COMPLETO DE NUESTRO GNU/LINUX.

4) Algo muy importante: Firestarter permite hacer que nuestro Linux funcione como proxy configurando los valores NAT de forma automática, sin tener que teclear valores y configuraciones (si no necesitamos esta función, podemos seguir adelante sin problemas).

5) Las reglas: Firestarter nos permite ver un listado muy completo de reglas a definir por Iptables.

Luego de seleccionar las reglas, podemos guardar la configuración para luego activar el firewall.

OTRAS CONFIGURACIONES

Si no nos gusta lo que generó automáticamente, podemos seguir con las configuraciones en forma manual y desde la misma aplicación. Para realizar una configuración en otro momento, sólo tendremos que ejecutar Firestarter y, previo ingreso de la clave de root, aparecerá la consola del server, donde encontramos varias opciones:

★ **Cortafuegos**: permite detener, arrancar o iniciar el asistente luego de la configuración inicial.

★ **Editar**: permite modificar la configuración o las preferencias en forma manual, sin usar el asistente.

★ **Ver**: brinda la posibilidad de visualizar los accesos y otros registros de nuestro equipo.

★ **Accesos**: muy importante porque permite bloquear una IP de un host, un nombre de host o el tráfico total, y habilitar o bloquear puertos; un cóctel de configuraciones importantes e útiles.

★ **Reglas**: permite añadir reglas personalizadas y borrar alguna existente.

Desde la misma aplicación, contamos con un manual en línea para evacuar todas las dudas con respecto al proceso de configuración.

COMO EJECUTAR LA CONFIGURACION AL INICIO

Para que Firestarter sea cargado en el inicio de nuestro sistema, sólo debemos ir a **Editar/Preferencias/General** y marcar el modo de operación en el panel del sistema. Firestarter generará una entrada en nuestro **init.d** para activar al inicio la configuración realizada y proteger el sistema.

CERRAR EL FIREWALL

Esta operación puede realizarse de distintas maneras, y varía mucho según la forma en que corre Firestarter. Si corre como servicio, podemos parar el servidor como lo hacemos con cualquier servidor corriendo en nuestro sistema:

```
# /etc/rc.d/init.d/firestarter stop
```

O deteniendo a Ipatbles:

```
# /etc/rc.d/init.d/iptables stop
```

En forma gráfica es posible desactivarlo desde la consola '**Firestarter Server**'.

EN CONCLUSION...

Siempre que hablamos de un firewall, se nos vienen a la cabeza complejas reglas y más reglas. Este potente soft nos da la posibilidad de brindar seguridad en nuestro equipo o servidor sin necesidad de tener muchos conocimientos previos. Además, no realiza ninguna acción de bloqueo por su cuenta, sino que todo es ejecutado por nuestro querido ipatbles. ¿Quién dijo que no podemos hacer seguridad de forma simple? ★ **Ariel Corgatelli**

FIRESTARTER NO ES UN FIREWALL; SU FUNCION ES EDITAR Y ADMINISTRAR LAS REGLAS DEL FIREWALL QUE CORRE EN NUESTRO SISTEMA GNU/LINUX (IPTABLES).

EL REEMPLAZO DE XFREE86

X.org

EN EL MUNDO DE GNU/LINUX, LA PARTE GRAFICA ES MANEJADA POR UN SISTEMA CLIENTE/SERVIDOR LLAMADO XWINDOW. HAY VARIAS IMPLEMENTACIONES DE EL, ALGUNAS PROPIETARIAS Y OTRAS LIBRES. EN ESTA OCASION, LES PRESENTAMOS UNA NUEVA IMPLEMENTACION QUE POCO A POCO VA FORMANDO PARTE DE LAS DISTRIBUCIONES MAS POPULARES.

X.org es una implementación del sistema XWindow en código abierto y de distribución libre. Sus funciones principales son brindar soporte para los dispositivos de entrada y salida, como teclado, mouse, placa de video, monitor, etc. La última versión estable es la Xorg 6.8, que puede descargarse desde su sitio oficial, ubicado en **www.x.org**. Allí encontraremos siete paquetes necesarios en formato tar.gz2:

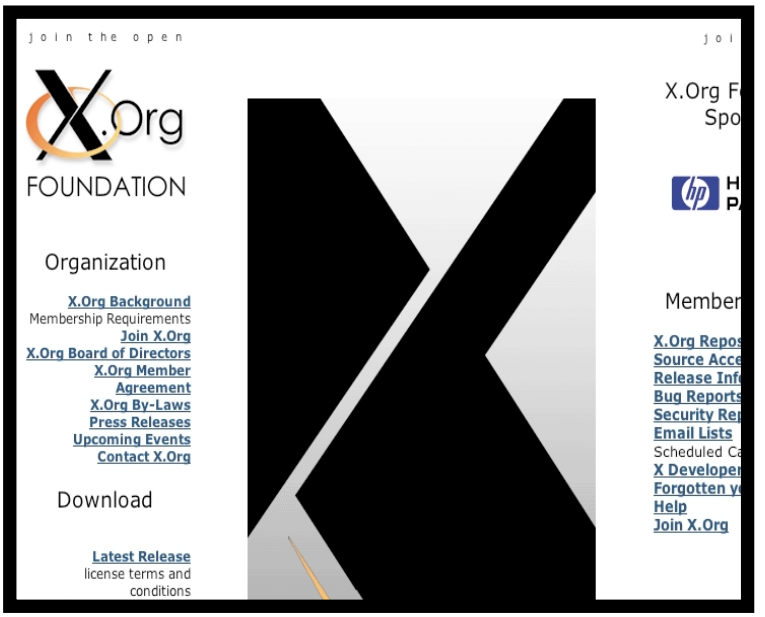
```
X11R6.8.0-src1.tar.gz
X11R6.8.0-src2.tar.gz
X11R6.8.0-src3.tar.gz
X11R6.8.0-src4.tar.gz
X11R6.8.0-src5.tar.gz
X11R6.8.0-src6.tar.gz
X11R6.8.0-src7.tar.gz
```

Las dependencias requeridas son:

```
libpng-1.2.5
expat-1.95.7
FreeType-2.1.7
Fontconfig-2.2.2
```

El primer paquete contiene las librerías básicas de X.org y varios programas de soporte; el segundo incluye programas X estándar; el tercero, el servidor X11 en sí; el cuarto y el quinto son destinados a las fuentes, y los últimos dos, a la documentación.
Una vez que los descargamos, es necesario comprobar la integridad de los paquetes (más aún si usamos módem), para lo cual tendremos que descargar el archivo **md5sums**. Luego, ingresamos el siguiente comando:

```
md5sum -c md5sums
```



EN EL SITIO OFICIAL DE X.ORG (WWW.X.ORG) ENCONTRARAN MAS INFORMACION SOBRE ESTE PROYECTO RELATIVAMENTE NUEVO.

La instalación es muy similar a la de Xfree86. Antes de proceder, hay que verificar que no tengamos instalado ningún servidor gráfico X. Luego desempaquetamos los siete paquetes X11 y creamos el directorio **/usr/X11R6** (si es que no contamos ya con él), que debe ser el directorio actual cuando ejecutemos (**preinst.sh**). **/usr/X11R6** deberá ser el directorio actual cuando ejecutemos ese script de preinstalación y se descompriman automáticamente todos los archivos. Luego podemos ejecutar el script de postinstalación **postinst.sh**.
Enlazamos el archivo **/usr/X11R6/bin/X** al servidor, que permite hacer la carga de nuestra tarjeta gráfica de la siguiente manera:

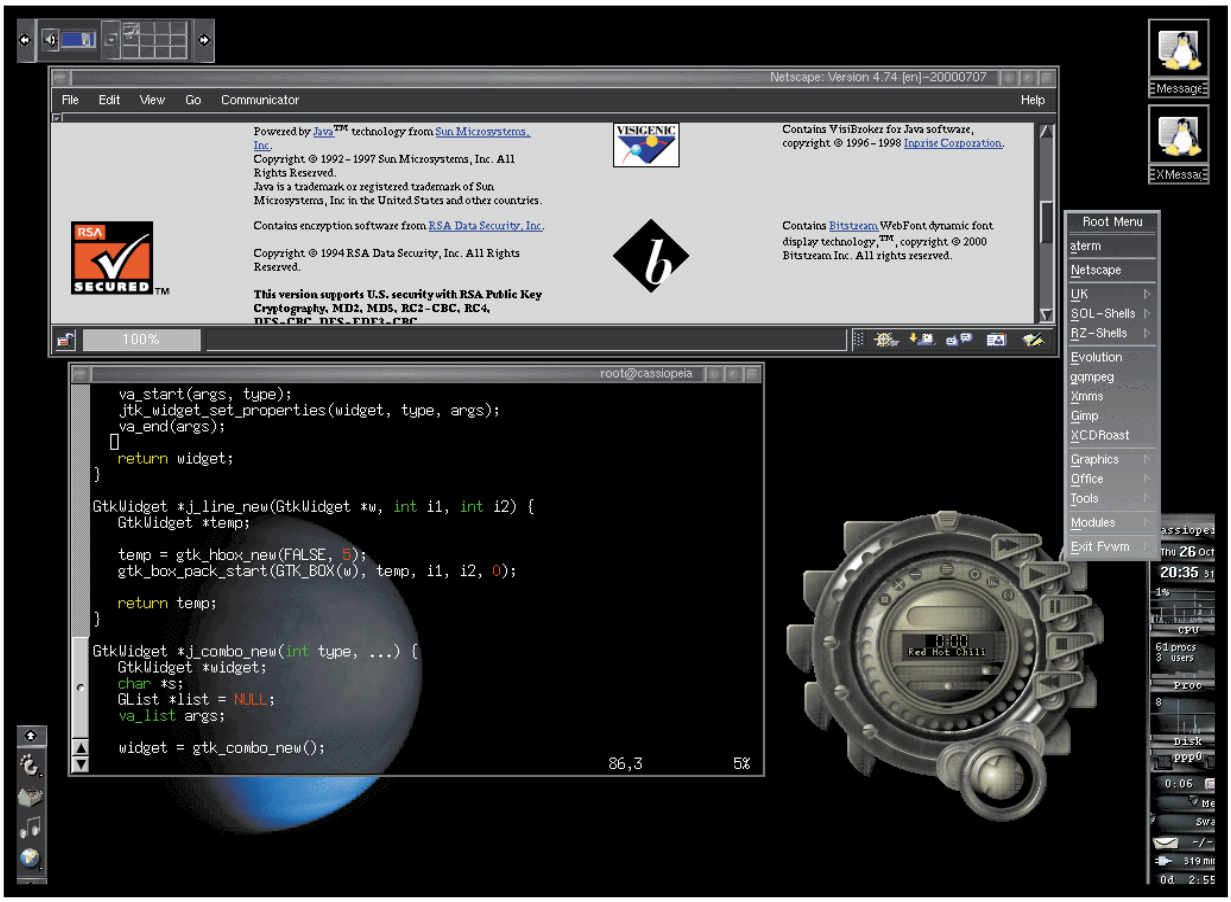
★ **Servidor de color de SVGA**: **/usr/bin/X11/X** debería estar enlazado con **/usr/X11R6/bin/XF86_SVGA**.

★ **Servidor monocromo**: para utilizar el servidor monocromo enlazamos X a **XF86_MONO**.

Lo mismo se aplica para el resto de servidores.
Luego, nos aseguramos de que el directorio **/usr/X11R6/bin** esté en el path, editando los valores predefinidos del sistema **/etc/profile**.
Finalmente, verificamos que **/usr/X11R6/lib** pueda ser localizado por **ld.so**, el enlazador en tiempo de ejecución. Para lograrlo, añadimos la siguiente línea al archivo **/etc/ld.so.conf**:

```
# cd /usr/X11R6
# sh /var/tmp/preinst.sh
```

Ejecutamos **/sbin/ldconfig** como root, y ahora sólo resta hacer la configuración.



LOS EFECTOS DE TRANSPARENCIAS SE VEN MUCHO MEJOR EN EL NUEVO X.ORG, TAL Y COMO PUEDE APRECIARSE EN ESTA CAPTURA DE PANTALLA DEL ESCRITORIO.

CONFIGURACION DEL SISTEMA

Para realizar este paso, comenzamos por crear el archivo **xorg.conf** con el siguiente comando:

```
# Xorg -configure
```

La pantalla se pondrá negra y tal vez oigamos algunos clics del monitor. Este comando creará el fichero **xorg.conf.new** en nuestro directorio personal. El próximo paso es editar el archivo **xorg.conf.new** para ajustar nuestro sistema. Los detalles del fichero se encuentran en la página de manual (**man xorg.conf**). Algunas secciones importantes son:

★ **Files**: cambia el orden de búsqueda de las rutas dedicadas a las fuentes. Es posible mover los tipos de fuentes o eliminar alguna.

★ **Monitor**: especifica los valores clásicos de refresco vertical y horizontal, si es que no fueron detectados automáticamente.

★ **InputDevice**: configuración de los dispositivos de entrada. Es posible cambiar la velocidad de autorrepetición del teclado agregando **Option "Autorepeat" "250 30"**.

★ **Device**: es una sección muy importante, ya que desde aquí es posible cambiar las opciones del controlador de video.

★ **Screen**: aquí se configura la resolución de la pantalla y la cantidad de colores.

Una vez que terminemos, tendremos que copiar el archivo a su ubicación final, mediante el comando **mv**:

```
mv xorg.conf.new /etc/X11/xorg.conf
```

Sólo resta comprobar si nuestro nuevo sistema funciona. Para acceder al servidor, ingresamos el comando **X** (debemos asegurarnos de que no esté apuntando a alguna versión de Xfree86 que tengamos instalada en el sistema).
De esta manera, sólo obtendremos una pantalla gris con un puntero del mouse en forma de X. Podemos cerrar la sesión presionando CTRL+ALT+RETR. También podemos acceder a los archivos log de X.org, ubicados en **/var/log/Xorg**.

EN CONCLUSION...

X.org es un proyecto de código abierto relativamente nuevo (en comparación con la historia de Xfree86). Aún hay muchas distribuciones que incluyen el viejo Xfree86, pero la tendencia es que todas comiencen a migrar al nuevo X.org. ¿Las razones? Varias: el desarrollo de X.org se lleva adelante con mucha más velocidad que el de Xfree86; las nuevas tecnologías (como las ventanas transparentes) se implementan más rápidamente, y los archivos de configuración están bastante más claros y ordenados. Si a esto le sumamos que el rendimiento de X.org se está optimizando al máximo, entonces ya no hay razones para no darle una oportunidad. ★ **Ariel Corgatelli**

COMPARATIVA DE REPRODUCTORES MULTIMEDIA

Reproductores multimedia

EN ESTA OPORTUNIDAD COMPARAMOS LOS MAS POPULARES REPRODUCTORES MULTIMEDIA. ¿HAY MUCHOS? ¿NO SABEN CUAL ELEGIR? EN ESTA NOTA LOS AYUDAREMOS A DECIDIR.

★ XINE

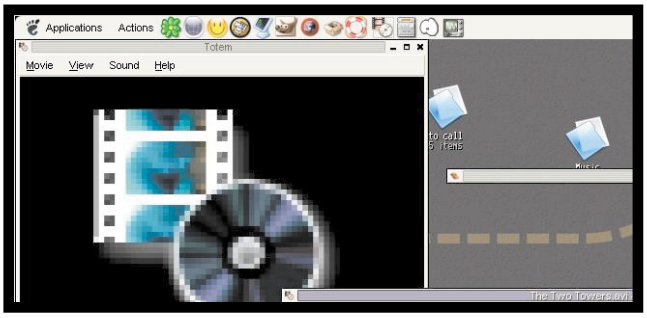
URL	www.xinehq.de
Versión actual	1.0

Reproduce cualquier medio, como Mplayer, pero Mplayer funciona más suave. Xine tiene un soporte completo y funcional para los DVDs, pero no puede cargar subtítulos como Mplayer (de forma externa al video). Se integra totalmente con el escritorio Gnome, con KDE e, incluso, con Mozilla, para permitirnos ver videos de la Web. Su última versión es xine 1.0.

★ TOTEM

URL	www.hadess.net/totem.php3
Versión actual	0.100

Este reproductor multimedia se incluye en casi todas las distribuciones comerciales. Soporta gran cantidad de formatos (WMV, MPEG, AVI, MP3, y otros) para ver DivX y DVDs, y escuchar MP3. Cuenta con una interfaz muy agradable, con la función adicional de capturar una pantalla en formato PNG. Su última versión es la 0.100.



★ KSCD

URL	www.kde.org
Versión actual	3.3.2

Es sólo un reproductor de audio, y está incluido en la mayoría de las distribuciones. Originario del entorno gráfico KDE, es posible utilizarlo en forma muy simple en los sistemas Linux gracias al supermount (montaje automático) de la lectora de CDs. El único requisito es que debemos contar con una lectora de CDs instalada para poder utilizarlo.

★ OGLE

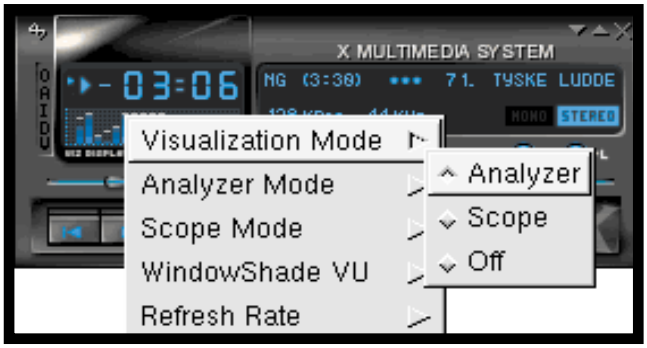
URL	www.dtek.chalmers.se/groups/dvd/index.shtml
Versión actual	0.9.2-1

No podía faltar un reproductor de DVDs en esta comparativa. Su funcionamiento es muy simple, al igual que su instalación. Este reproductor se ampara bajo la licencia GNU, y cuenta con varias versiones para distintos sistemas operativos: Solaris, Linux BSD, OpenBSD, FreeBSD, Linux, etc. Fue creado por un grupo de estudiantes de la Universidad Tecnológica de Chalmres, y es recomendado por ser uno de los más livianos, sin dejar de lado las buenas prestaciones que ofrece. Su última versión disponible es la v0.9.2-1.

★ XMMS

URL	www.xmms.org
Versión actual	1.2.10

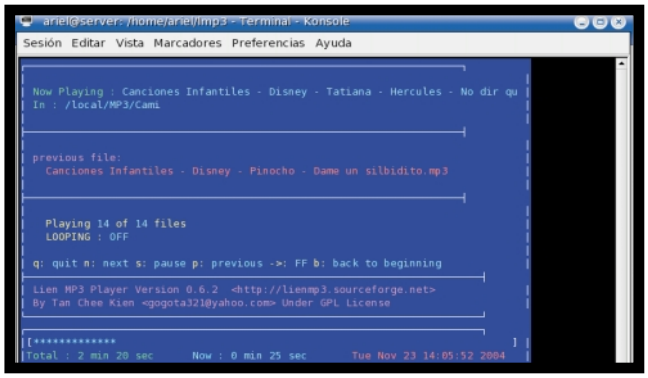
X MultiMedia System (XMMS) es un reproductor para varios formatos de audio, con una interfaz configurable basada en X/GTK+. Si buscamos un clon del popular Winamp, ésta es la verdadera opción. Es capaz de reproducir Audio Mpeg Layer 1,2 y 3 (con el plugin mpg123), Ogg Vorbis (con el plugin de Ogg Vorbis), CD-Audio no conectado a la tarjeta de sonido (con el plugin cd-audio) con soporte de CDDB, WAV, RAW, AU, MOD, XM, S3M, y otros. Además, cuenta con un buen número de plugins de efectos sobre la salida de audio.



★ LMP3

URL	http://lienmp3.sourceforge.net
Versión actual	0.6.2

También es posible escuchar música en formato MP3 Wav en una consola de texto. ¿Para qué? Para consumir menos hardware, por ejemplo. El programador recomienda tener un equipo con un mínimo requerimiento: procesador Pentium 90 MHz y, en lo referente al software, GNU Library 2.0.
El paquete completo comprimido tiene un tamaño de 100 k, viene listo para descomprimir y hacer el Make. Su última versión es la lmp3-0.6.2.



★ MPLAYER

URL	www.mplayerhq.hu
Versión actual	1.0

Podríamos decir que éste es el reproductor multimedia con más funciones para nuestro querido sistema GNU/Linux. Soporta la mayoría de los formatos, como MPEG, VOB, AVI OGG/OGM, VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLI, RM, NuppelVideo, YUV4MPEG, FILM, RoQ y PVA.
Otra notable característica de MPlayer es el amplio rango de controladores de salida soportados. Es posible la visualización en pantalla completa, y también cuenta con la posibilidad de ingresar subtítulos. Su última versión es la 1.0pre5.

★ GQMPEG

URL	http://gqmpeg.sourceforge.net
Versión actual	0.91.1

Este programa reproductor de MP3 desde sus comienzos optó por alejarse de la interfaz de Winamp y utilizar características propias. Permite la reproducción de varios formatos de sonido digital. Su interfaz proporciona mayor detalle en lo que respecta a la información del archivo que se va a reproducir, permite observar el uso de la CPU (recursos del sistema) y presenta accesos fáciles para llamar al editor de la lista de reproducción y las opciones. Su última versión es gqmpeg-0.91.1.

CONCLUSION

En sus comienzos, GNU/Linux no se caracterizaba por ser de lo más amigable en el ambiente multimedia, pero hoy en día, las cosas están cambiando, y cada vez más la comunidad está desarrollando software libre con tendencias multimedia. En esta comparativa pudimos ver como plato fuerte a XMMS, el reproductor por excelencia para los que vienen de Windows; Mplayer se ubica como el mejor reproductor de vídeo; en tanto que Gqmpeg le brinda pelea a Xmms en el sector audio, por ser muy original en todos sus aspectos. ★ Ariel Corgatelli

OPTI

¿YA INSTALARON GNU/LINUX? ¿ANDA BIEN? NOS ALEGRAMOS, PERO... PODRIA ANDAR MUCHO MEJOR. LAS DISTRIBUIDORAS DE GNU/LINUX GENERALMENTE SELECCIONAN LAS APLICACIONES Y LAS ARMAN A SU GUSTO, NO AL NUESTRO. LA CONSECUENCIA: EL SISTEMA NO ESTA FUNCIONANDO REALMENTE AL 100%. POR ESO, EN ESTE NUMERO DE LINUX USERS LES PRESENTAMOS UNA SERIE DE CONSEJOS Y TRUCOS QUE LES PERMITIRAN OPTIMIZAR EL RENDIMIENTO DEL SISTEMA, TANTO A NIVEL FUNCIONALIDAD COMO SEGURIDAD. ¡PREPAREN SUS EDITORES DE TEXTO Y MANOS A LA OBRA!

Juan Marcelo Rodríguez

¿Por qué hace falta optimizar GNU/Linux?

En realidad, esto depende de la distribución que tengamos. Generalmente, las distribuciones vienen compiladas para un procesador i386, i586 o i686, es decir que si nosotros tenemos un Pentium 4, no estaremos aprovechando al máximo la PC. Además, muchas distribuciones vienen con diversos servicios configurados para que inicien al momento del booteo. Cosas como éstas hacen que nuestro GNU/Linux consuma más recursos, al margen de que, dependiendo de los servicios que se inicien, quizá el sistema sea más inestable e inseguro. ¿Por qué inseguro? Bueno, porque estos servicios abrirán distintos puertos a nuestra red o a Internet, lo cual nos volverá más vulnerables a los ataques.

Por todo lo expuesto, en esta nota trataremos de guiarlos para que puedan configurar su GNU/Linux de manera óptima. Habiendo hecho la presentación pertinente, nos disponemos a poner manos a la obra.

Las mejores distribuciones de GNU/Linux para hacer tuning

Dentro del mundo de GNU/Linux hay muchas distribuciones, algunas de las cuales nos permiten afinar el sistema de forma más fácil que otras.

Debemos decidir qué distribución nos conviene más: una fácil de instalar y configurar, con todos sus paquetes compilados para que funcionen en la mayoría de las PCs del mercado; o una que, si bien es difícil de instalar y configurar, nos da la libertad de modificar el sistema completo y de compilar cada uno de sus paquetes. Dentro de este último tipo, están Debian (**www.debian.org**), Gentoo (**www.gentoo.org**) y Slackware (**www.slackware.org**), entre otras. Estas distribuciones son de las más documentadas y modificadas a lo largo de la historia.

Si bien Gentoo Linux es relativamente nueva en comparación con las otras dos, es espectacular al momento de querer optimizar nuestro GNU/Linux, ya que podemos compilar absolutamente todos los paquetes que queramos, utilizando las opciones y flags que deseemos (mientras nuestro hard y el programa lo soporte; si no, nunca compilará). Además, a diferencia de las otras dos, podemos hacer esta tarea de manera práctica y sencilla gracias a un sistema llamado Portage, que automáticamente descarga de Internet el código fuente del paquete que queremos compilar, busca sus dependencias (librerías u otros programas que necesite este paquete para funcionar o para poder compilarlo), baja los sources y los compila. De esta forma, podremos tener absolutamente todos los paquetes compilados a nuestro gusto y optimizados para nuestro tipo de procesador específico.

En la página de Gentoo encontraremos documentación en varios idiomas (incluyendo el español) para aprender a usarlo, además de los foros en donde podremos plantear preguntas o encontrar nuevos métodos para optimizar el sistema.

Pero como dije antes, no todo es color de rosa, porque lo que ganamos en velocidad de

IMIZACIÓN

EN EL SITIO OFICIAL DE GENTOO (WWW.GENTOO.ORG) ENCONTRARAN MUCHA INFORMACION ACTUALIZADA SOBRE ESTA DISTRIBUCION.

DEBIAN NOS DEMUESTRA SU SENCILLEZ DESDE LA PANTALLA DE BIENVENIDA. LUEGO, PODEMOS LLEVARLA HASTA DONDE QUERAMOS.

procesamiento en los programas al tenerlos compilados para nuestra PC, lo perdemos compilando cada uno de ellos. Para tener un sistema básico con KDE se demora aproximadamente 20 horas entre la creación del sistema de archivos y el último paquete compilado, en un Athlon XP 2000+ con 256 MB de RAM. Además, el sistema se debe configurar íntegramente en modo texto, y sólo con un editor no tendremos utilidades como **linuxconf**, **drakconf** o **yast** que nos simplifiquen la vida.

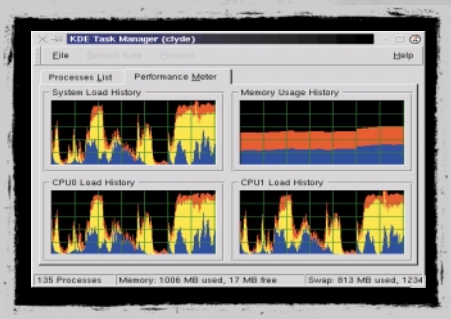
Con respecto a su instalación, ocurre lo mismo: es complicada y en modo texto, aunque se espera que para finales del primer trimestre de 2005, se libere un instalador gráfico para Gentoo (ya hay algunos no oficiales, como el de la distribución VidaLinux, **desktop.vidalinux.com**).

La otra opción que recomendamos es Debian, que, a diferencia de Gentoo, ya trae una gran cantidad de paquetes compilados. Debian incluye un sistema de instalación en modo texto no demasiado sofisticado, pero suficiente. En cuanto a la configuración, prácticamente no trae herramientas visuales ni autodetección de hardware, por lo que también nos obliga a configurar todo a mano con la ayuda de un simple editor de textos. Una de las cosas realmente interesantes de Debian es el sistema de actualización de paquetes APT, que permite instalar y gestionar aplicaciones instaladas de forma totalmente sencilla. Finalmente, casi todos los paquetes de Debian están optimizados para funcionar en su máximo rendimiento.

Slackware, por su parte, es considerada la distribución de GNU/Linux más parecida al UNIX original. Como Debian y Gentoo, no incluye muchas herramientas de configuración automática y nos obliga a configurar todo a mano. Su punto flojo es que aún utiliza un sistema de empaquetado muy "rústico": ¡el viejo **.tar.gz**!

De más está decir que las distribuciones que autodetectan todo, que preguntan poco y que configuran todo automáticamente no son ideales para hacer tuning, ya que nos darán mucho más trabajo que distribuciones como Debian, Gentoo y Slackware, que, al no tener prácticamente nada preconfigurado, nos dejan el camino libre para definir lo que queramos, en vez de tener que eliminar decenas de configuraciones que nos son inútiles, para luego configurar el sistema a nuestro gusto.

Midiendo el rendimiento



CUANDO LA COSA SE PONE MUY COLORADA, ES SIGNO DE QUE DEBEMOS ALIVIANAR LA CARGA PARA LA CPU Y LA RAM, CON EL FIN DE QUE NUESTRA POBRE COMPUTADORA NO SUFRA DEMASIADO.

Por lo general, el único que sabe exactamente cómo funciona el sistema es el propio usuario. ¿Quién no se da cuenta cuando la computadora se vuelve una carreta al tratar de abrir una imagen? Para evitar esta situación, lo mejor es conocer en qué momentos ocurre. De más está decir que si el sistema está siempre lento (es decir, desde que inicia hasta que lo apagamos), entonces lo mejor será cambiar directamente de distribución o hacer un arduo trabajo de afinación. Pero cuando el sistema se vuelve lento en determinadas ocasiones, lo más adecuado es recurrir a alguna herramienta que nos indique cuándo sucede. Para esto, disponemos del típico comando **top**, que nos informa en pantalla, entre otras cosas, cómo está la carga de la CPU y de la memoria RAM, cuánto espacio libre hay, y otros datos.

Si prefieren algo más visual, Ktop (**www.suse.de/~cs/ktop**) es, posiblemente, la herramienta adecuada, porque informa el rendimiento del sistema en tiempo real y mediante gráficos. Los usuarios de GNOME encontrarán en la barra de tareas algunas herramientas que cumplen funciones similares.

Configuración de servicios

Según qué distribución tengamos, habrá varios servicios (programas que corren en segundo plano para realizar distintos tipos de tareas) que se ejecutarán al inicio del sistema. Muchos de ellos no serán de uso cotidiano, y además de consumir recursos, pueden afectar su estabilidad y seguridad. Lo mejor será, entonces, dejar sólo aquellos servicios que usemos habitualmente. Por ejemplo, si no estamos en una red o no compartimos nuestros archivos con una red Windows, podemos desactivar el servidor Samba; luego, si llegamos a necesitarlo, lo único que deberemos hacer será iniciarlo.

Hay varias maneras de desactivar los servicios, según nuestra distribución. Por ejemplo, en Mandrake Linux podemos hacerlo desde DrakXServices, una utilidad del paquete DrakConf. En SuSE, usaremos la utilidad YAST, dentro de la cual debemos ir a Servicios de Red. Por su parte, en RedHat o Fedora iremos a linuxconf.

Para iniciar los servicios en forma manual, escribimos en la línea de comandos lo siguiente:

```
# /etc/initd/nombredelservicio start
```

Para detenerlos:

```
# /etc/initd/nombredelservicio stop
```

Y para reiniciarlos:

```
# /etc/initd/nombredelservicio restart
```

Otra forma es utilizar **services-admin** de Gnome. Los servicios que tienen un tilde a su izquierda son los que se inician por defecto. Para parar un servicio, hacemos clic en el botón Propiedades del servicio y elegimos la opción Iniciar; para pararlo, hacemos lo mismo pero escogemos Detener.

Miren bien la lista de servicios que se ejecutan al inicio y detengan aquellos que no suelen usar, como Samba, cliente dhcp, servidor web, servidor MySQL, etc.

Recuerden que los servicios son programas que se ejecutan en segundo plano, por lo que consumen tiempo de procesador y parte de la memoria, además de que pueden hacer vulnerable nuestra computadora a ataques o cuelgues.

Utilizando sólo los servicios que necesitamos, ahorraremos recursos del sistema y estaremos más seguros. Cuando queramos iniciar algún servicio especial, podremos usar alguna de las herramientas antes nombradas o, simplemente, recurrir a la línea de comandos.

Optimización de las aplicaciones

Como explicamos al principio de este artículo, todo archivo que no fue compilado en nuestra computadora seguramente fue compilado en otra que no tenía una configuración similar a la nuestra. Esto hace que el archivo binario que estemos usando no sea 100% óptimo en nuestro sistema. ¿La solución? Comenzar a compilar los programas más importantes que utilicemos. En general, el proceso suele ser complicado, pero con la siguiente guía podrán tener las herramientas básicas.

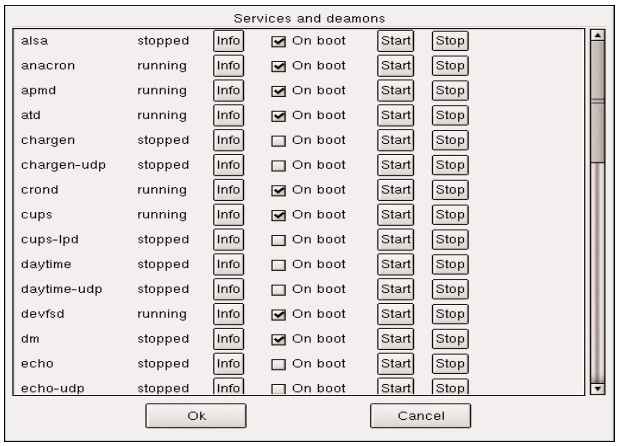
Asegúrense de descargar los programas para compilar en formato .tar.gz, que generalmente corresponde a código fuente. Una vez que lo hayan hecho, deberán copiarlo a /usr/src y descomprimirlo con el siguiente comando:

```
tar –zxvf archivo.tar.gz
```

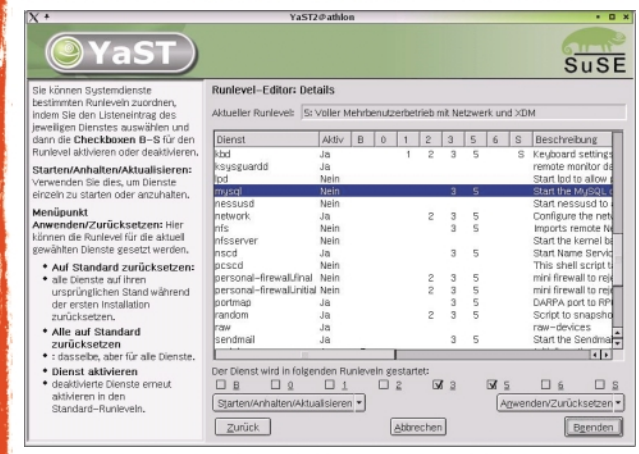
Suponiendo que bajamos un programa llamado abiword-2.2.0, el comando sería:

```
tar –zxvf abiword-2.2.0.tar.gz
```

Esto generará dentro de /usr/src un directorio llamado **abiword-2.2.0**, al cual debemos entrar. Luego, el primer comando que debemos tipear es **./configure** (sí, con el ./). Este comando se



EL SISTEMA DE CONFIGURACION DE SERVICIOS DE MANDRAKE (DRAKXSERVICES) ES MUY SENCILLO DE USAR.



YAST (INCLUIDO EN SUSE) INCLUYE UN EDITOR DE NIVELES DE EJECUCION QUE NOS PERMITE DEFINIR QUE SERVICIOS SE EJECUTARAN EN CADA NIVEL.

encarga de identificar nuestro sistema y sus componentes, y de armar un archivo Makefile que le indicará al compilador las opciones óptimas de compilación en cada caso. Si nos falta alguna librería o algún componente esencial para el funcionamiento del programa que queremos compilar, entonces **configure** nos dará un error, y deberemos ir a buscar ese componente para instalarlo siguiendo el mismo procedimiento que estamos describiendo.
Si el comando configure finalizó correctamente, entonces luego podemos ingresar **make** (sin parámetros), que se encargará de compilar el código fuente. Cuando finaliza, **make install** copia el binario a su directorio correspondiente, y cada uno de los demás archivos también a su lugar. Entonces, ya podremos ejecutar nuestra flamante aplicación compilada en el sistema y optimizada para que funcione lo mejor posible en nuestro hardware.

Configurar el kernel según nuestro hardware

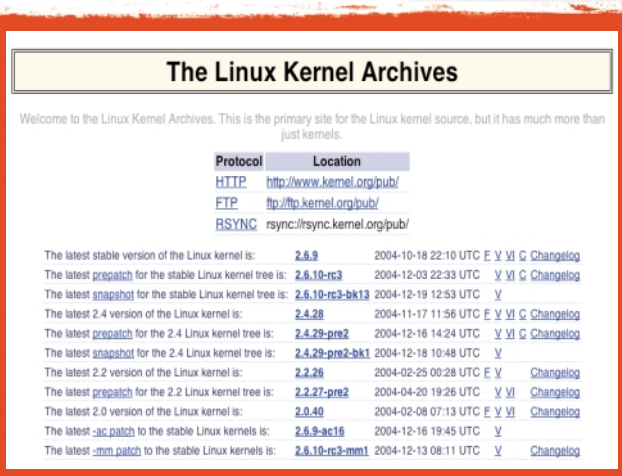
Como todos ya saben, el kernel es el núcleo de nuestro sistema GNU/Linux. Si está mal configurado, pueden pasar varias cosas, desde que nuestro sistema sea inestable o le falte soporte para nuestro hardware, hasta que directamente no funcione. Por eso es importante conocer nuestro hardware antes de configurar y compilar el kernel.
En las distribuciones comerciales, tales como SuSE, Mandrake y Fedora, el kernel viene configurado para todos los dispositivos que puede soportar y compilado para funcionar en una arquitectura compatible con la mayoría, como por ejemplo, i586. Esto se hace para que la misma distribución sea compatible con un amplio rango de computadoras y para que el usuario inexperto no tenga que aprender a compilar el kernel con el fin de que soporte algún dispositivo.
Lo que debemos hacer nosotros es configurar qué tipo de procesador tenemos exactamente y quitar el soporte a los dispositivos que no haya en nuestra PC. En el número anterior de Linux USERS tratamos el tema de la compilación y la configuración del kernel. En resumidas palabras, la configuración se hace desde el código fuente del kernel (que se puede descargar de **www.kernel.org**), con el comando make xconfig desde el modo gráfico (les conviene leer la documentación incluida en el paquete para obtener una explicación más detallada).
El soporte para nuestros dispositivos se puede configurar de dos formas: embebiendo el controlador (driver) de nuestro dispositivo dentro del kernel o configurándolo como módulo.
A nosotros nos conviene dejar el kernel lo más liviano posible; por lo tanto, si podemos configurar el controlador como módulo, deberemos hacerlo, ya que así ayudaremos a la estabilidad del sistema.
Los módulos deben cargarse ya sea al inicio del sistema o cuando los necesitamos, y puede hacerse desde la línea de comando con:

```
# modprobe [modulo]
```

donde [modulo] es el nombre que corresponda. Por ejemplo, para cargar el módulo de nvidia ingresamos:

```
# modprobe nvidia
```

Para aprender a configurar el kernel es aconsejable leer **www.insflug.org/COMOs/Kernel-Como/Kernel-Como.html**.
A nivel seguridad, otra es la historia. Si bien debemos dejar el kernel lo



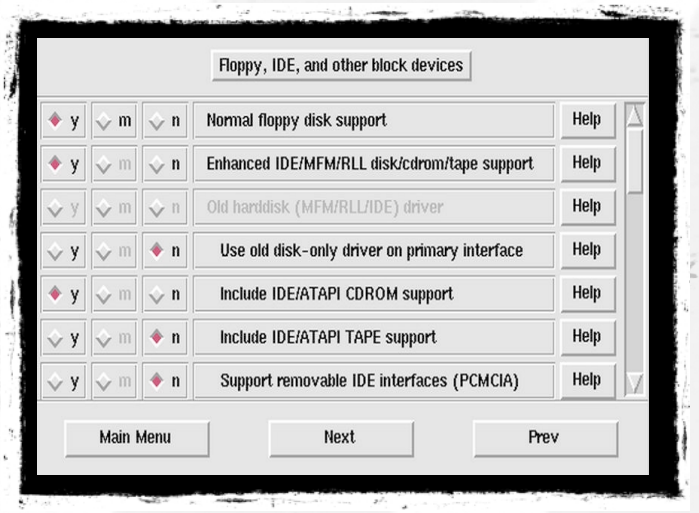
The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just kernels.

Protocol	Location
HTTP	http://www.kernel.org/pub/
FTP	ftp://ftp.kernel.org/pub/
RSYNC	rsync://rsync.kernel.org/pub/

The latest stable version of the Linux kernel is:	2.6.9	2004-10-18 22:10 UTC	F V VI C Changelog
The latest prepatch for the stable Linux kernel tree is:	2.6.10-rc3	2004-12-03 22:33 UTC	V VI C Changelog
The latest snapshot for the stable Linux kernel tree is:	2.6.10-rc3-bk13	2004-12-19 12:53 UTC	V
The latest 2.4 version of the Linux kernel is:	2.4.28	2004-11-17 11:56 UTC	F V VI C Changelog
The latest prepatch for the 2.4 Linux kernel tree is:	2.4.29-pre2	2004-12-16 14:24 UTC	V VI C Changelog
The latest snapshot for the 2.4 Linux kernel tree is:	2.4.29-pre2-bk1	2004-12-18 10:48 UTC	V
The latest 2.2 version of the Linux kernel is:	2.2.26	2004-02-25 00:28 UTC	F V Changelog
The latest prepatch for the 2.2 Linux kernel tree is:	2.2.27-pre2	2004-04-20 19:26 UTC	V VI Changelog
The latest 2.0 version of the Linux kernel is:	2.0.40	2004-02-08 07:13 UTC	F V VI Changelog
The latest -ac patch to the stable Linux kernels is:	2.6.9-ac16	2004-12-16 19:45 UTC	V
The latest -mm patch to the stable Linux kernels is:	2.6.10-rc3-mm1	2004-12-13 08:11 UTC	V Changelog

EL SITIO OFICIAL DEL NUCLEO LINUX (WWW.KERNEL.ORG) FUE RENOVADO, Y AHORA ES UN POCO MAS SOFISTICADO, AUNQUE MANTIENE SU SIMPLEZA. SI QUIEREN BAJAR LA VERSION COMPLETA DEL CODIGO FUENTE, RECUERDEN PRESIONAR SOBRE EL ENLACE "F".



CUANDO CONFIGURAMOS EL KERNEL, PODEMOS SELECCIONAR QUE COMPONENTES SERAN INCLUIDOS DENTRO DE EL, CUALES NO Y CUALES DIRECTAMENTE NO NOS INTERESAN.

más liviano posible y con la menor cantidad de controladores para evitar problemas que pueda haber en el código de éstos (que influirían en la estabilidad de nuestro sistema), nos conviene embeber los módulos y deshabilitar el soporte para cargarlos. De esta forma, a un atacante se le complicará cargar algún módulo malicioso a nuestro kernel. Si bien esto es práctico a nivel seguridad (para un server sería lo mejor), a nivel usuario de escritorio no es conveniente, ya que, por ejemplo, los controladores de Nvidia necesitan ser cargados al kernel como módulos.

Optimizar el desempeño de nuestros discos IDE

Una tarea importante que debemos hacer es verificar el buen desempeño de nuestros discos rígidos, utilizando el programa hdparm, que podemos descargar de **freshmeat.net/projects/hdparm**. Antes de empezar, debemos tener en cuenta que un mal uso de esta herramienta puede hacernos perder datos o, según la documentación, dañar los discos. El comando **hdparm** nos permite visualizar y modificar los parámetros de los discos duros IDE. Para ver qué parámetros tenemos activados, escribimos:

```
# hdparm /dev/hda
```

En mi computadora, por ejemplo, se muestra lo siguiente:

```
/dev/hda:
 multcount = 16 (on)
 IO_support = 1 (32-bit)
 unmaskirq = 1 (on)
 using_dma = 1 (on)
 keepsettings = 0 (off)
 readonly = 0 (off)
 readahead = 256 (on)
 geometry = 59323/16/63, sectors = 30616363008, start = 0
```

Para visualizar los parámetros que soporta el disco duro, ponemos en la línea de comandos:

```
# hdparm -i /dev/hda
```

En mi PC el resultado es:

```
/dev/hda:

 Model=SAMSUNG SV3063H, FwRev=PL100-21, SerialNo=0287J1BR303249
 Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
 RawCHS=16383/16/63, TrkSize=34902, SectSize=554, ECCbytes=4
 BuffType=DualPortCache, BuffSize=426kB, MaxMultSect=16, MultSect=16
 CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=59797584
 IORDY=yes, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
 PIO modes: pio0 pio1 pio2 pio3 pio4
 DMA modes: mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5
 AdvancedPM=no WriteCache=enabled
 Drive conforms to: ATA/ATAPI-4 T13 1153D revision 17:

 * signifies the current active mode
```

Si queremos conocer la velocidad de transferencia del disco, tenemos que cerrar la mayoría de los programas y servidores que estén corriendo, además del modo gráfico, y ejecutar:

```
# hdparm -Tt /dev/hda
```

En mi computadora esto da:

```
/dev/hda:
 Timing cached reads: 936 MB in 2.00 seconds = 467.84 MB/sec
 Timing buffered disk reads: 76 MB in 3.01 seconds = 25.25 MB/sec
```

Al habilitar todos los parámetros posibles de nuestro disco, obtendremos una velocidad óptima. Por ejemplo, para habilitar el acceso UDMA al disco debemos poner:

```
# hdparm -d1 /dev/hda
```

Para obtener más información acerca de cómo utilizar este comando, es conveniente leer la ayuda, con:

```
# hdparm –help
```

O ir a su página de manual, con:

```
# man hdparm
```

ANTES DE EMPEZAR, DEBEMOS TENER EN CUENTA QUE UN MAL USO DE HDPARM PUEDE HACERNOS PERDER DATOS O, SEGUN LA DOCUMENTACION, DAÑAR LOS DISCOS. !!!

Importancia del usuario root

Si bien esto no tiene directamente que ver con la optimización de nuestro sistema, es importante destacarlo: el usuario root sólo debe emplearse para tareas administrativas del sistema, como la configuración; pero para las tareas cotidianas, como leer mails, navegar por la Web, escuchar música, etc., deberíamos emplear un usuario que no tuviera privilegios de administrador; generalmente basta con que pertenezca al grupo users. De esta manera, protegemos nuestro sistema frente a ataques ya sea de virus como de código malicioso; nada ni nadie que use nuestra cuenta podrá llegar a tocar más que nuestro directorio home.
El usuario root es el usuario más importante del sistema. Si un atacante obtiene acceso a él, la única forma de volver a confiar otra vez en el sistema es reinstalarlo. Para evitar esta situación, hay ciertas reglas que debemos tener en cuenta:

ALGUNAS DISTRIBUCIONES, COMO UTUTO-E, NOS ADVIERTEN DEL PELIGRO DE UTILIZAR EL USUARIO ROOT CON PAPELES TAPIZ COMO EL DE LA IMAGEN.

Siempre crear un usuario para el trabajo diario. Si éste necesita cada tanto el acceso a root, agregarlo al grupo 'wheel'. Esto hace posible que un usuario normal recurra al comando "su" para tener temporalmente los privilegios de administrador.

Nunca debemos correr X como usuario root, éste debe ser usado sólo para tareas administrativas. Si existiera una vulnerabilidad en la aplicación y la estuviéramos corriendo como usuario, un atacante sólo podría ganar acceso a nivel usuario; mientras si estuviéramos con root, podría hacer lo que quisiera con nuestro sistema.

Siempre debemos usar las rutas completas a las utilidades que queremos ejecutar como usuario root (o siempre que usemos "su" –éste reemplaza nuestras variables de entorno por las del usuario root–, debemos fijarnos que el path de root contenga directorios protegidos, como /bin y /sbin). De no hacer esto, es posible para un atacante trucar el path para ejecutar una aplicación diferente de la que nosotros queremos, pero si el path es protegido o usamos rutas absolutas, esto no pasará nunca.

Si un usuario debe correr un par de comandos como root, en vez de darle todos los privilegios de superusuario, podríamos utilizar sudo, aunque hay que tener cuidado también con esto.

Nunca debemos dejar una terminal abierta cuando estamos conectados como root.

Escaneo de nuestros puertos abiertos

Nmap es una utilidad desarrollada para escanear puertos abiertos. > >
Dependiendo de cuáles estén en esta situación, estaremos más vulnerables a ataques; por lo tanto, es bueno no tener muchos o no tener puertos abiertos innecesariamente. Para descargar Nmap, vamos a la página www.insecure.org, donde encontraremos información acerca de su uso (para consultar el manual en español, la página es **www.insecure.org/nmap/nmap-fingerprinting-article-mx.html**).
Si queremos ver qué puertos tenemos abiertos, sólo debemos ejecutar:

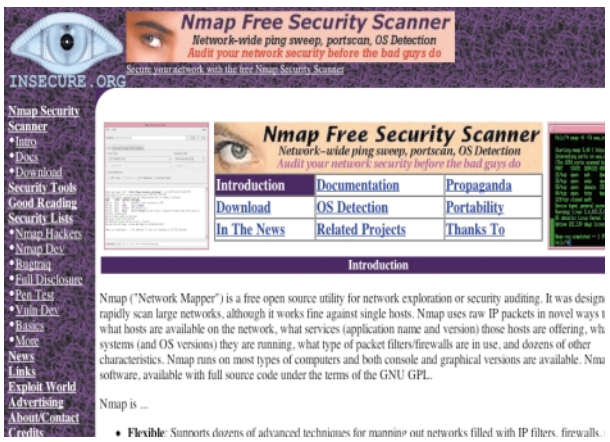
```
# nmap [ip]
```

En mi equipo, este escaneo da:

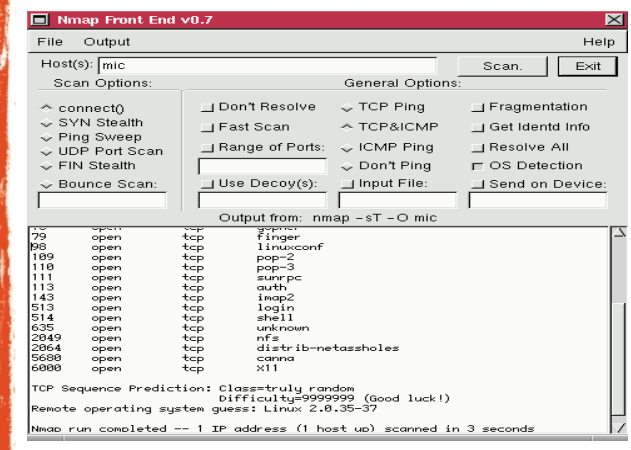
```
# nmap Q
Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2004-12-13 18:19 ART
All 1663 scanned ports on Q (192.168.1.2) are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 0.247 seconds
```

Si, por ejemplo, tuviéramos corriendo el servicio ssh (Secure Shell), se vería así:

```
#nmap Q2

Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2004-12-13 18:36 ART
Interesting ports on Q2 (192.168.1.5):
(The 1662 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
```



NMAP ES UNA HERRAMIENTA IDEAL PARA VERIFICAR QUE SERVICIOS ESTAN SIENDO PROVISTOS AL EXTERIOR. PUEDEN DESCARGARLA DE WWW.INSECURE.ORG/NMAP.



NMAPFE ES UNA INTERFAZ VISUAL PARA NMAP. PUEDEN ACCEDER A TODAS SUS OPCIONES DE MANERA RAPIDA Y SENCILLA.

```
22/tcp open ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 0.688 seconds
```

Con Samba activado, veremos algo como lo siguiente:

```
Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2004-12-13 18:39 ART
Interesting ports on Q (192.168.1.2):
(The 1661 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
139/tcp open netbios-ssn
445/tcp open microsoft-ds

Nmap run completed -- 1 IP address (1 host up) scanned in 0.251 seconds
```

Si no solemos utilizar algún puerto de los que aparece en la lista que vemos con nmap, lo mejor será dar de baja el servicio.

Configuracion del Servidor X

Por defecto, XFree suele estar configurado para actuar como un servidor X. Esto puede ser muy peligroso ya que utiliza conexiones no encriptadas por TCP y queda a la espera de clientes X. Si nosotros no necesitamos esto, debemos desactivarlo, para lo cual podemos iniciar X utilizando el comando:

```
# startx -- -nolisten tcp
```

Si queremos hacerlo en forma permanente, debemos configurar el archivo "/usr/X11R6/bin/startx" y agregarle la siguiente línea:

```
defaultserverargs="-nolisten tcp"
```

En caso de usar un login gráfico, debemos hacer lo mismo pero de forma diferente:

* Si usamos GDM (Gnome Display Manager), debemos editar el archivo **/etc/X11/gdm/gdm.conf** y modificar la línea que dice:

```
[server-Standard]
command=/usr/X11R6/bin/X
```

por:

```
[server-Standard]
command=/usr/X11R6/bin/X -nolisten tcp
```

* Si usamos xdm (X display Manager) o kdm (Kde Display Manager) debemos editar el archivo "/etc/X11/xdm/Xservers" y cambiar:

```
:0 local /usr/bin/X11/X
```

por:

```
:0 local /usr/bin/X11/X -nolisten tcp
```

Por último, si tenemos una placa de video Nvidia o ATI, debemos bajar unos controladores especiales (de código fuente cerrado) que nos permitirán usar aceleración 3D; esto lo hacemos desde:

NVIDIA: **www.nvidia.com/object/linux.html**
ATI: **www.ati.com/support/driver.html**

Si nuestra distribución es SuSE, podemos utilizar la herramienta YOU (Yast Online Updater) para descargar e instalar estos drivers (necesitaremos privilegios root).
Ambos paquetes incluyen información acerca de la manera de instalarlos.

Consejos de Buanzo para optimizar servidores

➽ En principio, toda distribución basada en código fuente, como Gentoo o Sourcemage, es óptima en ambientes que no requieren correr aplicaciones que deban ser certificadas, más por cuestiones comerciales que técnicas, como Oracle. Si no existe este impedimento, dichas distribuciones pueden optimizarse para el hardware en el que corren. En este punto hay un arma de doble filo: dependiendo de los parámetros que se definan para el compilador GNU C, los binarios funcionarán sólo para la arquitectura especificada, ya que el compilador, mediante el parámetro march, sabe qué instrucciones son válidas. Lo más útil sería utilizar el parámetro "mcpu" o la combinación "march" / "mtune". La página del manual de gcc explica en detalle las posibles incompatibilidades y el propósito de cada parámetro de optimización vinculado a la arquitectura. Por otra parte, en el ámbito de la red corporativa no es recomendable filtrar todo el protocolo ICMP, pero sí ciertos subtipos (dependiendo de las necesidades, el RFC 792 explica todo en detalle). Pueden revisar un listado de RFCs útiles en **www.networksorcery.com/enp/protocol/icmp.htm**, o todos los RFC en existencia en **www.rfc-editor.org**). Si un servidor emite un paquete que termina siendo filtrado por DROP, este servidor dejará un socket esperando una respuesta, la cual nunca llegará. Con REJECT, obtiene la respuesta y no se mantiene una lista de estados sin sentido. Otro ejemplo es el de ubredes y routers mal configurados, donde, tal vez, con el tiempo aparezca una tabla de rutas tan grande, que ralentizará todo el servidor. En el caso del kernel Linux, no se va a colgar, pero en otros casos... hay que tener un especialista en "reset".

Consejos de Lorenzutti para optimizar la seguridad

➽ Cuando comencemos una instalación de un servidor, a la hora de seleccionar los paquetes, debemos elegir lo mínimo indispensable. Paquetes de más y cosas que no sabemos qué hacen ni para qué sirven son un riesgo que tenemos que evitar. Lo mejor es saber todo lo que está corriendo, utilizando un simple "ps".

➽ Armar un esquema controlado de particiones, donde nada pueda crecer indefinidamente. Como norma básica, los directorios que puedan crecer en función del uso del equipo deberían estar en una partición separada; leáse: /tmp, /var/log y /home, como para empezar a hablar. Luego, considerando el uso final del equipo, nos convendrá armar particiones para el servicio en concreto. Por ejemplo, si estamos armando un servidor Apache, cae de maduro que debemos dejar en una partición aparte el directorio /srv/www/. Para armar algo realmente poderoso sería perfecto aprender a manejar LVM o EVMS.

➽ Tener sólo activos los servicios que están configurados correctamente. Debemos verificar que no estemos iniciando servicios que no necesitemos. Para esto, con un simple "netstat -tanp | grep LISTEN" podremos ver qué puertos tenemos abiertos y en qué direcciones de red. ¿Es necesario tener corriendo un ssh en el puerto 2222? No hace falta que hagan un nmap desde fuera; comprueben con un "netst..".

➽ Siempre estar al día con los parches de seguridad. Lamentablemente, todo el tiempo hay noticias de nuevos exploits. No hay otra opción más que estar controlando estas cosas, y nadie mejor que las listas de seguridad de la distribución que estén usando para mantenerse al tanto de las novedades.

Consejos de Arena para optimizar las estaciones de trabajo

➽ Si tienen una máquina del diablo, entonces no estarán muy interesados en optimizar su rendimiento, porque todo irá a la velocidad del rayo frente a sus ojos. La cuestión es qué hacer cuando el hardware es más bien humilde, y las aplicaciones parecen comer los recursos, como el viejo Pacman se comía las semillitas en el laberinto. OK, en ese caso, les recomiendo hacer lo siguiente:

➽ Busquen herramientas que permitan hacer lo mismo pero usando menos recursos. Siempre las hay. ¿OpenOffice se arrastra? Reemplácenlo por Abiword y GNUMeric, que con 16 MB de RAM andan a las mil maravillas. Busquen, y siempre van a encontrar.

➽ ¡Quiten todos los servicios del inicio que no utilicen!

➽ No utilicen KDE ni GNOME si no les da la RAM. ¡Hay muchos manejadores de ventanas que son muy lindos y consumen mucho menos! Denle, una oportunidad a IceWM, WindowMaker, BlackBox, FVWM, XFCE…

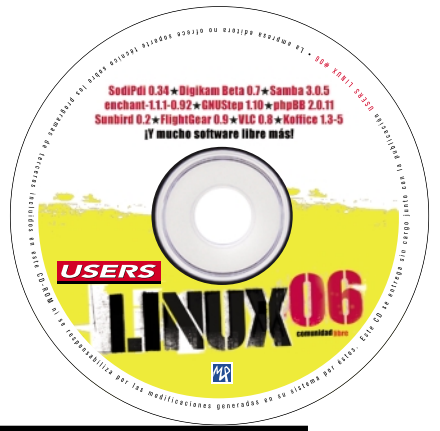
➽ Si tienen las agallas, reemplacen Xfree86 por X11.Org. Está mucho más optimizado.

➽ Mantengan siempre particiones SWAP que sean el doble de la memoria RAM. Si agregan RAM, entonces agranden la SWAP.

➽ Si sus ojos no distinguen entre miles y millones de colores, entonces bajen la intensidad de color en la configuración de X, de 24 a 16. En sistemas de bajos recursos la diferencia de rendimiento es increíble.

LES CONTAMOS LO MEJOR QUE INCLUIMOS EN EL CD DE ESTE MES

EN ESTA OPORTUNIDAD TRATAREMOS DE ELEGIR LAS DIEZ MEJORES APLICACIONES QUE INCLUIMOS EN EL CD, DAREMOS UNA BREVE DESCRIPCION DE CADA UNA Y SELECCIONAREMOS LA MAS DESTACADA.



En el CD

Como lo venimos haciendo en ediciones anteriores, éste es un resumen del mejor Software Libre incluido en el CD #06.
Este mes tratamos de presentar las novedades de fines de 2004 y principios de 2005.
Por suerte, el 2004 fue un año repleto de sorpresas en cuanto a Software Libre, y como integrantes de este maravilloso mundo GNU/Linux, no dejamos nada afuera.
También aprovecho para que nos hagan llegar sugerencias con respecto al soft que les gustaría ver incluido en los CDs (tanto al correo de la redacción como al mío personal, **arielcorg@tectimes.com**).

BOTIQUIN

En esta categoría, como siempre, tratamos de ofrecer todo el software necesario para poder resolver dependencias de aplicaciones incluidas en el CD.
La estrella de esta edición es X.org, en su versión 6.8, con todas las fuentes necesarias para llevar a cabo la instalación de este nuevo servidor gráfico.
X.org está reemplazando a Xfree en la parte gráfica (lo mismo ocurrió en el CD #04 con ALSA y OSS). Para entrar en más detalles, también tenemos un artículo completo sobre el tema en esta publicación.

Xorg 6.8
www.x.org
Licencia: Open Source

EDITORES

Una selección especial de editores web, de programación y de texto en general. Como no podía ser de otra manera, elegimos la última edición de NVU, que siempre nos asombra con sus nuevas versiones de editores web.
En este caso, presentamos NVU 6.0 con GTK incluido, un útil editor web que nos permite trabajar en un sitio web en forma dinámica, al mejor estilo de Macromedia.
Como desventaja, podemos decir que no cuenta con una versión en español; como ventaja, todas sus funciones están muy bien logradas y son muy fáciles de usar.

NVU 0.6
www.nvu.com
Licencia: GNU/GPL

LA ULTIMA VERSION DE THE GIMP.

HOGAR

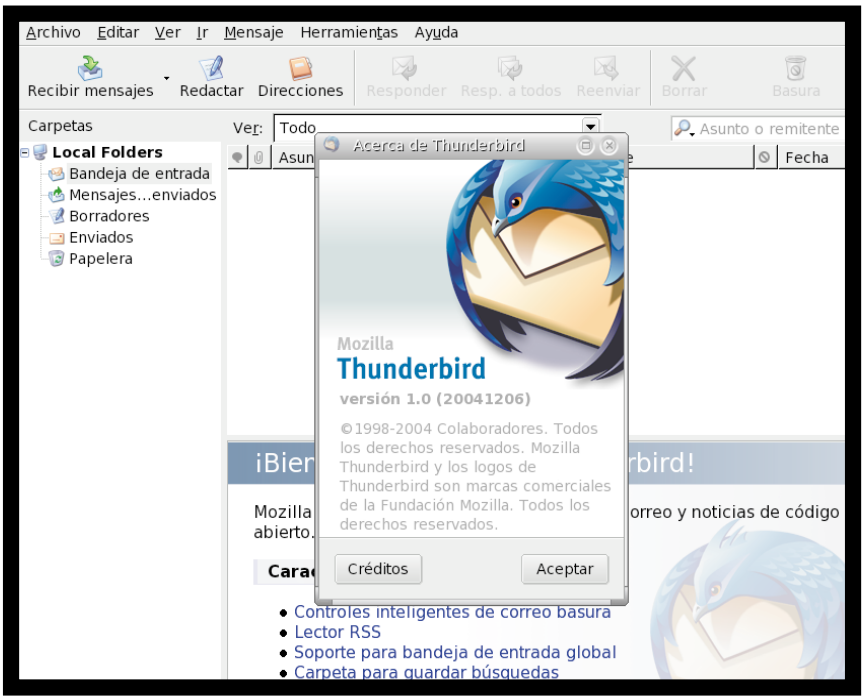
En esta categoría, nos dedicamos a cubrir todas las necesidades de los usuarios hogareños de computadoras. Este mes incluimos las librerías para utilizar GMail con un cliente de correo electrónico, una aplicación para manejar cámaras digitales, una base de datos de fuentes clásicas y, como estrella, la última versión del popular GIMP 2.2. Para los que no lo conocen, es la mejor opción en su rubro para editar, retocar, crear, modificar y trucar imágenes. Gimp es altamente recomendable, y puede compararse con el paquete de Adobe (Photoshop). También cuenta con la posibilidad de convertir formatos de imagen.

Gimp 2.2
www.gimp.org
Licencia: GNU/GPL

INTERNET

En esta edición, todos los méritos se los lleva la gente del proyecto Mozilla, que nos viene sorprendiendo versión tras versión, tal como ustedes pueden palpitar desde nuestras publicaciones donde les contamos las novedades. A final de 2004 vieron la luz las últimas versiones de Thunderbird 1.0 y Firefox 1.0 en español, y elegimos nada menos que a Thunderbird como el soft destacado de esta categoría. Se trata de uno de los clientes de correo electrónico más usados en el mundo de GNU/Linux, ya que soporta todo tipo de servidores y permite importar datos de clientes anteriores. Cuenta con capacidades mejoradas de temas y actualizaciones, y soporta múltiples cuentas con manejo totalmente transparente de ellas. La instalación es mucho más simple que la descarga: sólo hay que desempaquetarlo y buscar el binario.

Thunderbird 1.0 Español
www.mozilla.org
Licencia: Open Source



THUNDEBIRD 1.0, EL FIREFOX DE LOS CLIENTES DE CORREO ELECTRONICO.

JUEGOS

Si bien Linux no se caracteriza por ser de lo mejor en cuanto a desempeño de juegos, nosotros tratamos de mostrar que también se puede jugar con nuestro querido pingüino. En esta oportunidad destacamos a Doom3. Ahora ya no tenemos que recurrir a los emuladores como Cedega para usarlo: en el CD incluimos la versión 3 ejecutable para Linux. Lo único malo que encontró nuestro experto en juegos, Juan Pablo Firrincielli, es que no soporta ALSA, así que tendremos que esperar un poco en cuanto al sonido.

Doom 3
zerowing.idsoftware.com/linux/doom
Licencia: Propietaria

MULTIMEDIA

En esta categoría, presentamos las últimas novedades y herramientas para la grabación de multimedia. Este mes elegimos una aplicación ya tratada en el primer número de USERS Linux: K3B en su versión 0.11.18. En esta edición, se mejora el 0soporte de nuevos dispositivos y se sigue con la política de facilitar la grabación de CDs y DVDs en GNU/Linux.

K3b 0.11.18
www.k3b.org
Licencia: GNU/GPL

OFICINA

En esta categoría incluimos todas las aplicaciones relacionadas con el Software Libre para uso ofimático, como procesadores de texto, planillas de cálculo, suites de oficina completas, programas contables, y mucho más.
Galopin es la estrella de este mes, por ser un programa de facturación y control de stock muy simple de usar, liviano, útil y, por supuesto, en español. Es altamente configurable, no necesita muchos requisitos para funcionar y es por demás ideal para usar en las PyMEs.

Galopín 2 Beta
galopin.sinuh.org
Licencia: GNU/GPL

DESARROLLO

Todo el software de programación libre está en esta sección. Encontraremos este mes unas cuantas herramientas interesantes.
Phoenix 1.5-6 Object Basic es un entorno de programación visual bajo Linux totalmente orientado a objetos, mediante el cual los programadores que están acostumbrados a trabajar en Visual Basic podrán empezar a desarrollar bajo Linux. Al usar un diseño visual muy claro y un lenguaje base de programación como Basic, la curva de aprendizaje es muy suave.
También contamos con **Eclipse 3.0.1**, una plataforma para el desarrollo de herramientas de programación.

Phoenix 1.5-6
www.janus-software.com
Licencia: Open Source

SERVIDORES

Una categoría que abarca servidores y, también este mes, algunas utilidades muy importantes a la hora de administrar una red. En esta ocasión destacamos la última versión de Samba. Para los que no lo conocen, es el sistema que permite que máquinas Windows y máquinas Linux se comuniquen sin problemas. Es uno de los servidores más utilizados en una red, por lo cual es preciso estar al tanto de las vulnerabilidades y parches de las versiones liberadas.

Samba 3.0.5
us1.samba.org/samba
Licencia: GNU/GPL

UTILITARIOS

En este caso, encontramos una herramienta para el manejo de dispositivos, como mtools; otra para la descarga de archivos, Azureus; y la estrella: Firestarter. Esta aplicación es un manejador de reglas de firewall al mejor estilo Windows (aunque evita sus cuelgues y pantallas azules).

Firestarter 0.9.3
fs-security.com
Licencia: GNU/Gpl

CONCLUSION...

Como siempre, sólo quisimos resumir el software incluido en el CD y sus categorías. Quedó mucho material fuera, pero a nuestro entender, mencionamos lo más importante.
Todas las sugerencias serán bienvenidas a **arielcorg@tectimes.com**. Mucha suerte ¡y a probar el soft!
★ **Ariel Corgatelli**

SOLUCIONES Y TECNICAS PARA USUARIOS DE GNU/LINUX

trucos

EN ESTA OPORTUNIDAD, COMO PERSONALIZAR XDM, PASAR ARCHIVOS DE TEXTO DE WINDOWS A LINUX, MATAR PROCESOS DE FORMA SENCILLA, Y ALGUNOS TRUCOS QUE NOS ENVIO UN LECTOR. ¡LOS INVITAMOS A QUE USTEDES TAMBIEN NOS MANDEN LOS SUYOS!

MATAR PROCESOS

En ocasiones, tenemos la necesidad de matar algún proceso que está corriendo (porque nos molesta o por cualquier otra razón). Esto puede realizarse de varias maneras en nuestro sistema. Primero, debemos saber qué procesos se están ejecutando. Veamos un ejemplo concreto:

En nuestro sistema la aplicación Mozilla está colgada y queremos cerrarla porque, de forma gráfica, no responde. Para hacerlo, vamos a una consola y nos fijamos que esté corriendo:

```
[ariel@server ariel]$ ps -auxlgrep mozilla
ariel 16914 1.0 16.7 146836 40264 ? Sl Dec06 15:32 /mozilla-
1.7.2/mozilla-bin
ariel 20241 0.0 0.3 1916 736 pts/1 S+ 16:24 0:00 grep mozilla
```

Vemos dos líneas: una devuelve el valor de proceso con el cual está corriendo, y la otra muestra la consulta que nosotros ingresamos. Para cerrar Mozilla hacemos uso del comando **kill**, que se utiliza de dos maneras. La primera es haciendo referencia al proceso número **16914**:

```
# kill -9 16914
```

La segunda es mucho más simple y útil:

```
# killall mozilla
```

De este modo, se cerrará todo lo relacionado con Mozilla, debido al uso de killall.
Con esta última opción, podemos matar un proceso o una aplicación que esté corriendo, de forma simple y sin saber nada más que su nombre.

TRUCOS DE UN LECTOR

Estaba leyendo la revista; muy buena, realmente. Les mando un par de comandos que me resultan útiles en mi RedHat 7.2:

1) Es frecuente querer capturar el texto de una página web, pero hacerlo con el navegador gráfico da bastante trabajo, porque hay que seleccionar, copiar, pegar en otro editor, salvar, etc. Es mucho más fácil hacerlo con:

```
[gerardo@magallanes]$ lynx -dump tectimes.com >
textimes.txt
```

y... ¡listo el pollo!

2) Aunque los discos son cada vez más grandes, siempre necesitamos hacer lugar. Cuando el árbol de directorios es grande, resulta complicado decidir cuál queremos eliminar. Entonces, llega el shell al rescate :-)

```
[root@magallanes]# cd /home
[root@magallanes]# du -S lsort -n lmore
```

3) Por lo general, el espacio que necesitamos está en el directorio **home** de nuestro benemérito jefe. En ese caso, habrá que buscar espacio adicional por otro lado, por ejemplo, viendo qué paquetes tenemos instalados. Siempre que estemos usando paquetes rpm, podremos recurrir a:

```
[root@magallanes]# rpm -qa --queryformat
'%8{SIZE}\t{NAME}\n' l sort - k1,1n lmore
```

Así sabremos el tamaño y el nombre del paquete. Luego, con el clásico [root@magallanes]# rpm -e <paquete> obtendremos el espacio que necesitamos, sin molestar al jefecito :-)

Gerardo Díaz
gerardodiaz.tripod.com

ACENTOS MIGRADOS

Muchas veces nos entregan documentos creados en cualquier editor de Windows o DOS, y cuando los abrimos en los editores de GNU/Linux, nos encontramos con la sorpresa de que los acentos fueron cambiados por simbolitos raros. Esto se debe a que tanto Windows como Linux utilizan diferentes configuraciones de caracteres. Para solucionar este problema, vamos a utilizar el programa **recode** de la siguiente manera:

```
recode ibmpc:lat1 documento.txt
```

Si ahora vemos nuestro archivo, debe estar corregido. Por el contrario, si queremos trasladar o compartir un archivo creado con un editor de Linux, la forma de realizarlo deberá ser la siguiente:

```
unix2dos='recode lat1:ibmpc' documento.txt
```

Si es necesario utilizar continuamente diferentes formatos, podemos incluir en el archivo de configuración **/etc/profile** las siguientes líneas:

```
alias unix2dos='recode
lat1:ibmpc'
alias dos2unix='recode
ibmpc:lat1'
```

soluciones

BUSCANDO ARCHIVOS

Mientras usamos nuestro sistema, podemos vernos en la necesidad de buscar algún archivo alojado en un directorio específico. Esto implica que deberemos saber bien cómo se usa el comando **find**, tal como muestra el ejemplo:

```
find / -name "nombre"
```

El símbolo **/** hace referencia al directorio inicial de la búsqueda, y podremos indicar que el proceso se realice dentro de cualquier directorio, por supuesto, con los permisos necesarios para la lectura.
Luego, **-name** nos permite ingresar un nombre de archivo para buscar por ese patrón.

Linux College
Carreras Presenciales y a Distancia
Cursos de Verano
Carrera Linux y Seguridad Avanzada
Certificación UTN Delta / Linux College
4824-2050
informes@linuxcolle.com.ar
www.linuxcollege.com.ar
www.carreralinux.com.ar
Marcelo T. de Alvear 2247
Buenos Aires
En alianza con:
CIUA.COM Solution
www.ciua.com.ar
CWG Capacitación Web Group
www.capacitacionweb.com.ar
www.escuelachina.com.ar

PERSONALIZANDO XDM

La mayoría de los usuarios de Debian utilizan entornos gráficos livianos y no tan cargados. Como gestor de sesiones, lo que viene predefinido es XDM, ni hablar de usar GDM o KDM.
Sobre todo el aspecto visual, XDM concentra su configuración en dos archivos clásicos: **Xresources** y **Xsetup**, que podemos encontrar en **/etc/X11/xdm/**. **Xresources** se encarga de controlar el login, y **Xsetup** controla todo el resto de la sesión gráfica. Vamos a un ejemplo en la edición de ambos archivos: ¿quieren cambiar el mensaje de bienvenida? Editen las siguientes líneas:

```
xlogin*greeting: Bienvenido a
Debian GNU/Linux
xlogin*greetFont: -adobe-
helvetica-bold-o-normal--20-180-
100-100-p-138-iso8859-15
xlogin*greetColor: Blue3
```

Para cambiar los campos de texto:

```
xlogin*namePrompt: Usuario:
xlogin*promptFont: -adobe-
helvetica-bold-r-normal--18-140-
100-100-p-105-iso8859-15
xlogin*promptColor: darkgreen
```

Si quieren cambiar el mensaje de fallo por algo más divertido:

```
xlogin*fail: ¡¡¡Otra Vez, Otra Vez...!!!
xlogin*failFont: -adobe-helvetica
bold-r-normal--18-140-100-100-
p-105-iso8859-15
xlogin*failColor: red
*Foreground: black
*Background: blue
```

Para cambiar el logo de Debian:

```
xlogin*logoFileName: /usr/X11R
6/lib/X11/xdm/pixmaps/debian2.xpm
```

El archivo debe estar en formato .xpm. En la página de Debian podemos bajar logos desde el apartado Miscelánea. Con la utilidad **convert** del programa GIMP, es muy sencillo pasar estos logos en .png o .jpg a .xpm.
Para cambiar los parámetros de la ventana de login:

```
xlogin*font: -adobe-helvetica
medium-r-normal--18-140-100
100-
p-100-iso8859-15 FUENTE AL
TECLEAR
xlogin*borderWidth: 1 # ANCHO
DE LA LINEA DEL BORDE
xlogin*frameWidth: 5 # ANCHO
DEL MARCO
xlogin*innerFramesWidth: 2 #
ANCHO DEL MARCO; VENTANAS DE
USUARIO Y PASSWORD
xlogin*shdColor: grey30 # COLOR
DEL MARCO EXTERNO. BORDE
SUPERIOR E IZQUIERDO
xlogin*hiColor: grey90 # COLOR
DEL MARCO EXTERNO. BORDE
INFERIOR Y DERECHO
xlogin*background: skyblue10 #
COLOR VENTANA DEL LOGIN
xlogin*foreground: darkgreen #
COLOR DEL TEXTO A TECLEADO
xlogin*useShape: true # USAR
BORDE DEL MARCO
xlogin*logoPadding: 5
```

Luego, sólo resta configurar XDM para usar como servidor gráfico, escapando al truco de configuración visual aquí explicado.

RESPONDEMOS A LAS CONSULTAS DE LOS LECTORES

Consultas

EN ESTA SECCION VAMOS A RESPONDER A LAS PREGUNTAS QUE USTEDES NOS ENVIAN A NUESTRA DIRECCION DE MAIL, linuxusers@tectimes.com

MANDRAKE NO SE INSTALA

Queridos amigos de USERS:
Tengo una buena PC –un Atlhon XP 2700 en un mother Asrock de 333 de bus, una masa–, pero no puedo hacer andar el CD de Mandrake 10. Fui a cambiarlo a la editorial y me dieron otro, pero en cuanto inicio la instalación, me tira un error y no anda más. Hice un disquete de arranque, probé en modo texto, etc., pero no puedo avanzar más que un poco, y luego aparece ese error, **boot failed**.
He instalado Mandrake 8 sin problemas, pero ni el 9 ni éste funcionan.
Espero que puedan ayudarme... gracias.
★ **cesar**
★ **cesar142229@yahoo.com.ar**

Linux USERS: Lo mismo nos consultó Camila Soledad (**camilasoledad22@yahoo.com.ar**), así que la respuesta va para los dos. Sin lugar a dudas, se trata de un problema de compatibilidad de alguno de los componentes del sistema de inicio (kernel, init, o algún módulo) con su hardware. Como no tenemos mucha información, la única recomendación que les podemos dar es que intenten instalar alguna otra distribución de GNU/Linux (UTUTO-E, SuSE, etc.).

INSTALANDO CONECTIVA

Hola a todos los que hacen USERS. Primero quiero decirles que sus revistas están buenísimas. He aprendido mucho con las USERS Linux 0#1, 2 y 3, y a pesar de que todavía soy principiante, ya me defiendo bastante bien en Mandrake.
Mi problema surgió cuando bajé la distribución Conectiva 10.0 (CD1, CD2, CD3 y el Live CD 0.5). En los CDs 2 y 3 sólo encuentro RPMs, y no tengo la menor idea de cómo hacer la instalación. Hay archivos ejecutables, pero no abre ningún asistente para la instalación; encontré un archivo llamado **rawriter**, y el **readme** que tiene dice que necesita un archivo llamado diskio.dll en un disquete, pero no sé para qué sirve eso. Tampoco sé cómo bootear el CD. Los archivos que encontré en el CD 1 son los siguientes: loadlin, rawriter, rawriterwin, rundos, gzip y fips, fips20, loadlin20, rdev y restorrb, todos ejecutables dentro de la carpeta DOSUTILS. En las demás carpetas sólo están los directorios del sistema dev, home, etc, root, tmp, entre otros, y más RPMs. Por favor, ¿pueden ayudarme?
También aprovecho la ocasión para preguntarles si puedo subscribirme siendo paraguayo y viviendo en Pilar, y en caso de que sea posible, cómo puedo abonar.
Les agradezco mucho y los felicito.
★ **Luis Martín Montiel Tejera**
★ **septemberdusk@yahoo.com.ar**

Linux USERS: Luis, todas las distribuciones basadas en RedHat se instalan de la misma forma. Lo ideal es intentar hacer que bootee el CD (hay que asegurarse de que esta opción esté habilitada en el Setup del BIOS de la computadora). Si no bootea, es porque el CD está mal grabado. En cuanto a los archivos ejecutables, **rawriter** es el que te permite generar un disquete de booteo. En la raíz debe haber un directorio llamado images, que contiene imágenes de diferentes discos de inicio, según tus necesidades. Con el disco de inicio, se bootea, y después se puede iniciar el sistema de instalación del CD.

¿QUÉ SON ESAS PALABRAS?

Quisiera hacerles una consulta con respecto a las líneas de comando del shell que tienen la forma inicial:

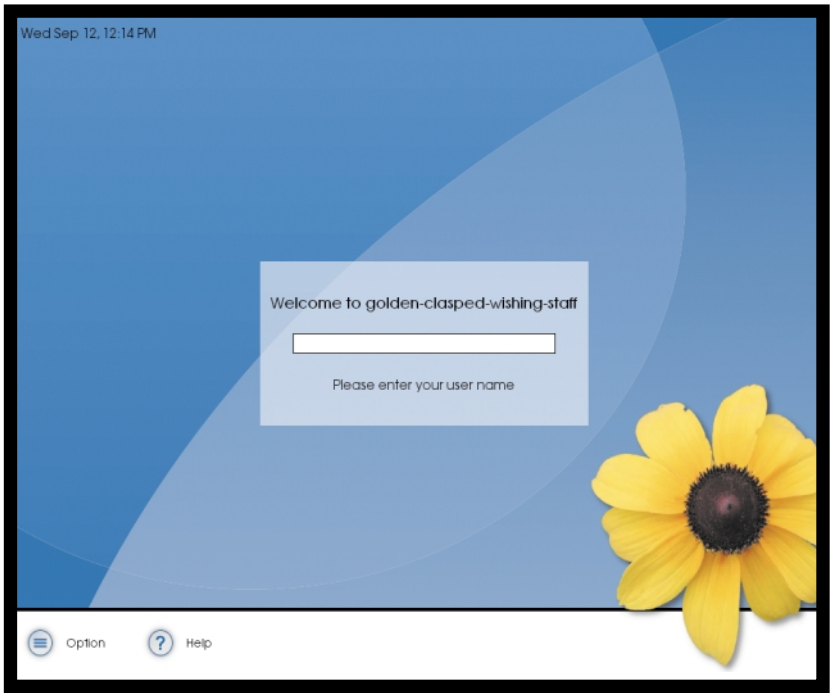
```
root@nixbox#
root@mindless#
root@braindless#
```

Recién estoy comenzando con Linux, y por problemas técnicos con mi PC, mi incursión es (por ahora) sólo teórica. El punto que quisiera aclarar es qué significan las palabras después del @, ya que busqué por todos lados y no encontré respuesta. Supongo que serán nombres de ejemplo, pero lo extraño es que en una misma nota los cambian. Si me pudieran contestar, les agradecería.

★ **Camaina Skay**
★ **CamainaSkay@hotmail.com**

Linux USERS: Amigo Camaina, lo que aparece después de la arroba es el nombre de la computadora (también conocido como nombre de host o hostname). Ese nombre lo elige el usuario cuando instala el sistema operativo, y después puede cambiarlo cuantas veces quiera. Entonces, el prompt quedaría así: **usuario@máquina.** Esto quiere decir que el usuario "usuario" está trabajando en la computadora "máquina". El "usuario" **está en** (arroba se dice "at" en inglés) "máquina".

CUANDO TENEMOS EL INICIO EN MODO GRAFICO, SE NOS INFORMA EL NOMBRE DE LA MAQUINA EN LA PANTALLA DE LOGIN.



PROBLEMAS CON KNOPPIX

Te escribo porque soy novata en Linux y quiero comenzar probando Knoppix. Tengo una Intel Pentium III con 128 MB RAM y 450 MHz, el motherboard es un PCCHIPS M748LMRT, y la placa de video es una SIS 620 Rev 2A. Lamentablemente, mi disco duro es de 9 GB, así que no puedo particionarlo para tener los dos sistemas operativos. Knoppix reconoce el hardware pero al final me sale un mensaje diciendo que no hay servidor X adecuado para la tarjeta; luego reintenta con el servidor Xfree86 (vesa), con el servidor xfree86(fbdev), etc. En los foros de TecTimes me dijeron que tengo problemas con la placa de video, pero que Knoppix tendría que funcionar con mi placa y que, de lo contrario, lo ejecute en modo texto, pero no sé hacerlo porque no conozco los comandos. Espero que puedas ayudarme y, desde ya, muchas gracias.

★ **María Eugenia**
★ **meugevaz@arnet.com.ar**

KNOPPIX ES LA DISTRIBUCION ELEGIDA POR MUCHOS USUARIOS CUANDO NO ES POSIBLE INSTALAR UNA DISTRIBUCION DE GNU/LINUX EN LA COMPUTADORA.

Facundo Arena: Knoppix suele tener problemas con las placas de video on-board. Pero la tuya tendría que funcionar sin problemas, y lo digo porque yo solía tener una igual. ¿La pantalla se queda negra? Si es así, quizá cargó el servidor pero no en la resolución correcta. Una opción es presionar la combinación de teclas CTRL ALT + y CTRL ALT - para cambiar la resolución; yo siempre tengo que hacerlo en Knoppix. Otra opción es tratar de conseguir alguna versión actualizada de Knoppix o de alguna otra distribución Live (como MandrakeMove), que posiblemente tenga los archivos mejor configurados.

PONIENDO LAS COSAS EN SU LUGAR

comando

El sistema UNIX nos ofrece varias herramientas para mantener la privacidad e integridad de nuestros datos. Una de ellas es el sistema de permisos. Con los permisos, podemos blindar nuestros datos y realizar así una protección que podríamos denominar: "seguridad informática de bajo nivel". ¿Por qué de bajo nivel? Muy simple: todos sabemos que para que un sistema sea considerado "seguro", deberá hacer uso de ciertas técnicas y herramientas como firewalls, seguridad en base a grupos de usuarios con sus correspondientes accesos, seguridad física de nuestro servidor, actualizaciones de bugs y parches de seguridad, compilaciones constantes para mejorar el kernel, backup de datos y, por último (si no me olvido de nada), podemos decir que también necesitamos asignar permisos en archivos y directorios. Algunas de estas herramientas, como la seguridad física (atornillar la computadora al piso para que no nos la roben) son consideradas de alto nivel. Las herramientas de seguridad que ofrece el kernel (como el sistema de permisos) se consideran de bajo nivel.

¿PARA QUE SIRVE ASIGNAR PERMISOS?

Esta pregunta se responde muy fácilmente. En un hogar, empresa o institución es muy normal que la información almacenada en las computadoras tenga algunas restricciones en cuanto a los usuarios que pueden acceder a ella, y en muchas empresas, esto mismo se puede dar por grupos. De esta forma, tenemos que llegar a manejar una política de permisos para poder proteger los datos frente a las miradas indiscretas de los demás usuarios y de seres ajenos a nuestro sistema.
No voy a entrar en detalles acerca de la manera de crear usuarios y grupos de usuarios; simplemente, quiero transmitirles la preocupación que deberemos tener en mente cuando nuestro equipo GNU/Linux esté funcionando.
Es de vital importancia, entonces, que cada uno tenga lo suyo y que tenga posibilidad de protegerlo para que nadie acceda a algo que no debe ver.

¿CUALES SON LOS PERMISOS?

GNU/Linux maneja el acceso a archivos y directorios sobre la base de permisos asignados, que son tres:

- **Lectura**: permite que los usuarios realicen lecturas del archivo.
- **Escritura**: permite modificar el archivo.
- **Ejecución**: permite que el usuario ejecute el archivo (siempre y cuando sea ejecutable, claro está).

Cuando se generan los permisos, el sistema guarda un listado de ellos, que luego puede ser visualizado utilizando comandos como **ls –l**.
Los permisos se identifican de forma muy simple, mediante las siguientes letras:

r: permiso de lectura.
w: permiso de escritura.
x: permiso de ejecución.

Para visualizar los permisos de los archivos, utilizamos el siguiente comando:

```
[ariel@server local]$ ls -l
total 272
drwxr-xr-- 2 ariel home 4096 nov 29 12:35 1/
-rwxr--r-- 1 ariel home 29341 nov 28 09:48
t.png*
drwxr-xr-x 2 ariel home 4096 nov 24 22:18 lug/
drwxrwx--- 3 ariel home 4096 sep 24 00:19
Cami/
```

La primera columna, con todas esas letras r, w y x desparramadas, nos indica los permisos que tiene ese archivo o directorio. Veamos la primera entrada (el directorio 1/).
La primera letra (**d**) nos indica que es un directorio. Luego, **rwx** nos dice que el propietario del archivo (usuario dueño) puede leer, escribir y ejecutar en ese directorio. Siguiendo de izquierda a derecha, **r-x** nos señala que los miembros del grupo del usuario pueden solamente leer y ejecutar en ese directorio (noten que no hay w, sino que está reemplazada por un símbolo ' –'). Finalmente, **r--** nos indica que los otros usuarios (es decir, los que no son dueños del archivo y no forman parte de su grupo) sólo pueden leer en dicho directorio.

La primera letra de la columna nos muestra el tipo de archivo. Tenemos una **'d'**, para indicar que no es un archivo, sino un directorio. Otros casos pueden ser:

chmod

EN NUESTRA SECCION DEDICADA A LOS PRINCIPIANTES, NO PODIA FALTAR ESTE TEMA TAN IMPORTANTE: EL SISTEMA DE PERMISOS EN ARCHIVOS Y DIRECTORIOS DE GNU/LINUX.

★(guión): al estar en blanco con sólo una raya, nos dice que es un archivo.
★**b**: nos muestra un bloque especial.
★**c**: nos muestra un archivo de carácter especial.
★**d**: un directorio.
★**l**: un enlace simbólico.

Ahora los nueve espacios siguientes nos muestran los permisos para el usuario, el grupo y el resto del mundo, como solemos decir en Linux.

ASIGNACION DE PERMISOS

Esta tarea puede realizarse de varias formas, pero nosotros veremos la manera de asignar permisos desde una terminal de texto (¡por eso estamos en la sección shell!). El comando que utilizaremos será **chmod**.
Este comando acepta tres modificadores para manejar permisos, definidos por los siguientes signos:

El signo - quita permisos.
El signo + agrega permisos.

Veamos algunos ejemplos. Con este comando agregamos permisos de lectura:

```
chmod +r [nombre de archivo]
```

Ahora veamos cómo quitar los permisos de escritura:

```
chmod -w [nombre archivo]
```

Y, finalmente, para asignar permisos de ejecución:

```
chmod +x nombre archivo
```

De esto se desprende que podemos asignar al mismo archivo los permisos de escritura, lectura y ejecución:

```
chmod +rwx [nombre de archivo]
```

Bien, éstos son permisos generales. Es decir, no especificamos a quién se los brindamos, por lo cual se brindan o quitan a todos. Si queremos ser más precisos, debemos definir a quién damos o quitamos el permiso. Para esto, utilizamos las siguientes letras:

u: usuario dueño del archivo.
g: grupo dueño del archivo.
o: otros usuarios.
a: todo el mundo.

Y ahora, para ser más específicos, veamos algunos ejemplos:

```
chmod a+w [archivo]
```

Con a+w todo el mundo puede escribir.

```
chmod u+x [archivo]
```

Con u+x el usuario dueño puede ejecutar el archivo.

```
chmod g-w [archivo]
```

Con g-w el grupo pierde los permisos de escritura.
Y, obviamente, podemos combinarlos:

```
chmod ug+wx
```

Con ug+wx el dueño y el grupo dueño del archivo pueden escribirlo y ejecutarlo.

EL MODO OCTAL

Para asignar permisos también contamos con el método octal. En la tabla podemos ver la forma.

000: equivale a ningún permiso.
001: permiso de ejecución para el propietario del archivo.
002: permiso de escritura para el propietario.
004: permiso de lectura para el propietario.
010: equivale a permiso de ejecución para el grupo.
020: permiso de escritura para el grupo.
040: permiso de lectura en el grupo.
100: ejecución para el resto del mundo.
200: escritura para el resto del mundo.
400: sólo lectura para el resto del mundo.

Entonces, podemos utilizar los números octales de manera muy simple; podemos sumarlos tomando como base la tabla anterior, y obtendremos como resultado un solo número por campo. Veamos algunos ejemplos:

```
chmod 777 [archivo]
```

Asigna todos los permisos para todos los usuarios.

```
chmod 740 [archivo]
```

En este caso, el usuario propietario tiene privilegios totales; el grupo, de sólo lectura, y el resto, nada.

EN CONCLUSION...

Cuando empezamos a utilizar GNU/Linux en forma cotidiana, nos damos cuenta de su potencial y de las posibilidades que tenemos para proteger nuestro sistema. Estas herramientas están siempre ahí para que las usemos de muchas maneras, y en este artículo sólo hemos presentado algunas. El sistema de permisos es una manera muy efectiva de proteger datos ante ataques y miradas indiscretas. Suerte y hasta la próxima. ★ **Ariel Corgatelli**

POTENCIANDO LAS CAPACIDADES DEL KERNEL 2.6

Aplicación de parches

EN UN ARTICULO DE USERS LINUX #04 HABLAMOS SOBRE LAS NOVEDADES DE LA SERIE 2.6 Y VIMOS UN MINI PASO A PASO PARA ACTUALIZARNOS. AHORA VAMOS A VER ALGUNOS PARCHES QUE HAY PARA LA VERSION 2.6 DEL KERNEL. MUCHAS DE LAS NOVEDADES DE ESTE KERNEL ESTUVIERON EN SU MOMENTO COMO PARCHES PARA LA SERIE 2.4. POR LO TANTO, ESTAREMOS INCLUYENDO EN UN KERNEL ACTUAL COSAS QUE TODAVIA NO SON PARTE DE LA VERSION ESTABLE, PERO QUE SEGURAMENTE LO SERAN EN BREVE.

Muchos se preguntarán qué es un parche del kernel. Bueno, es justamente de lo que estamos hablando. Usando como ejemplo un kernel 2.6.9, hay ciertas funcionalidades que no están incorporadas en la versión oficial, que está en **www.kernel.org**. Estos agregados, llamados "parches" son los que tendremos que utilizar para incluir las funcionalidades faltantes, por ejemplo, soporte para hardware más actual o propietario, como los drivers de NVIDIA.
Habiendo hecho esta aclaración, ahora tendríamos que ver alguno de los parches sugeridos para nuestro kernel de turno. Todos los parches que veremos tienen su página oficial, que figura al final del artículo en la lista de links sugeridos. Pero vamos a recomendar ahora un sitio en particular que está dedicado a este tema, un sitio donde hay un parche que incluye a todos los mencionados: freshmeat.net/projects/linux-cko (y ya que están por ver el link, no dejen de revisar todo freshmeat.net, porque, sin lugar a dudas, es "el" sitio para enterarse de las actualizaciones y novedades para soft en sistemas Open Source).

¿COMO APLICAMOS EL PARCHE?

¡La parte fácil! Bueno, primero hay que conseguirlo. La página sugerida ya la conocen, y al momento de escribir la nota, la versión para el kernel estable 2.6.9 es la "cko3". Les dejo la ruta completa al archivo, para que puedan hacer una descarga directa: kem.p.lodz.pl/~peter/cko/patch-2.6.9-cko3.bz2.
Acto seguido, tenemos también las fuentes del kernel 2.6.9 de **www.kernel.org** (si están en la Argentina, les sugiero el mirror **www.ar.kernel.org**, notablemente más rápido).
Con estos dos archivos, nos vamos al directorio correspondiente (/usr/src) y descomprimimos las fuentes del kernel y el parche:

```
# tar xjvfp linux-2.6.9.tar.bz2
# ln -s /usr/src/linux-2.6.9 /usr/src/linux
# bunzip patch-2.6.9-cko3.bz2
# patch -p0 <patch-2.6.9-cko3
```

El fundamento del link es porque el parche quiere escribir en /usr/src/linux, y el resultado de descomprimir las fuentes no es el directorio esperado.
Ahora podemos seguir leyendo el artículo, porque ya tenemos las fuentes del kernel con todo lo necesario, e iremos encontrando lo que vayamos nombrando a la hora de configurarlas.

SUPERMOUNT-NG

Este parche es para los que están agotados de montar y desmontar cdroms y floppys, para los que no quieren tener que volver a ver el mensaje de que el dispositivo se encuentra ocupado, para los que quieren que las unidades se comporten de forma parecida a... Windows. No los voy a engañar, me molesta mucho tener que montar y desmontar mi Pendrive USB cada vez que quiero usarlo. No me sirve mucho el método tradicional de trabajar con él, dado que la escritura en mi Pendrive tiene que ser sincrónica, así que la caché que me puede proporcionar no me ayuda demasiado.
En mi estación de trabajo necesito que mi Pendrive trabaje como en Windows: insertarlo y, sin tocar nada, leer; terminar de trabajar y que se desmonte. ¿Les pasa algo parecido? ¡Sigan leyendo!
La página principal del proyecto es **supermount-ng.sourceforge.net**, pero si siguieron la recomendación que les hice antes, pueden usar el parche general que se encuentra en el link ya mencionado. Una aclaración, en la página del proyecto todavía están un tanto atrasados con los releases para los kernels actuales. No obstante, puede ser que el parche no se aplique bien con la versión actual de kernel. No se lo vuelvo a repetir: trabajen con el parche recomendado.
Parchado el kernel, la opción en las fuentes se encuentra en la parte de sistemas de archivos, en la subcategoría de "Pseudo Filesystems".

Ahí hay un sistema de archivos nuevo llamado supermount. Sólo queda compilar el kernel, instalarlo y reiniciar. Esto va más allá del artículo, así que los invito a ver cómo empezar a utilizarlo luego de haber reiniciado.
Las modificaciones van a ser en el fstab. Ahí deberíamos tener algo así:

```
# egrep '(floppylcdrom)' /etc/fstab
/dev/cdrom   /media/cdrom    iso9660        ro,user,noauto
   0      0
/dev/fd0     /media/floppy   auto           ro,user,noauto      0
     0
```

Luego de los cambios, debería quedar de la siguiente manera:

```
# egrep '(floppylcdrom)' /etc/fstab
none       /media/cdrom     supermount
 defaults,fs=iso9660,dev=/dev/cdrom,--,ro   0       2
none       /media/floppy    supermount
defaults,fs=auto,dev=/dev/fd0,--,rw   0 2
```

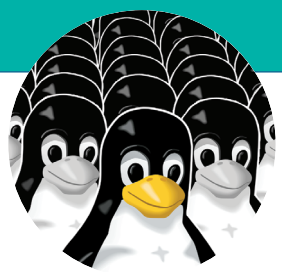
Presten atención, acabo de sacar el "**noauto**", y esto quiere decir que cuando reiniciamos, va a intentar montar estas declaraciones. Otra aclaración importante es que todo parámetro que figure luego del "--" que hay en la parte de opciones se pasa al sistema de archivos real, y no al supermount. Todo lo previo al "--" son las opciones del supermount; entre ellas, se le avisa el dispositivo qué debe controlar y el sistema de archivos por utilizar.
Pero cuidado: para aplicar estos cambios no hace falta reiniciar; hacemos un "mount -a" y luego ya podemos probar a hacer un mount, y veremos que si bien no tenemos ningún floppy o CD-ROM, figuran como montados. Ahora hacemos la primera prueba: insertamos un floppy y vamos al directorio correspondiente. Como por arte de magia, el directorio mostrará su contenido. OK, esto no parece un gran desafío dado que ya existía el "automount". Sacamos el floppy y, rápidamente, ponemos otro, y más rápido aún, volvemos a hacer un "ls". ¿Ven? Se percató del cambio de floppys, volvió a leer la unidad y nos mostró su nuevo contenido sin "montar y desmontar".

¿Qué sabés de Linux?

ESCUELA VIRTUAL
Cursos a distancia
www.linuxadistancia.com

CARRERA LINUX '05
9 materias / 8 meses
¡Abierta la inscripción!

CURSOS SIMPLES
Nivel Básico / Administración de Redes
Programación PERL



tuxsys
Soluciones libres para un mundo libre

Un emprendimiento de Héctor Facundo Arena

Informes e inscripción www.tuxsys.com.ar

Marcelo T. de Alvear 768 - 2º C ▪ Ciudad de Buenos Aires
Tel.: (54-11) 4313-9977 ▪ e-mail: info@tuxsys.com.ar
Horario de atención: lunes a viernes de 17 a 21 hs.

Publicidad sólo válida para la República Argentina.

¡No descorchen todavía! ¿La prueba final? Juguemos con los CD-ROMs: ahora podemos ejecutar uno estando parados en el directorio (debería decir que el dispositivo está en uso). Obviamente, también podemos cambiar de CD sin salir del directorio. Fantástico, nuestras unidades removibles se portan como en... coff... coff... Windows.

SHELL FILE SYSTEM

El SHFS es un sistema de archivos remoto que utiliza el servidor de SSH para acceder a un sistema de archivos. Es algo cómodo y simple si tenemos que intercambiar archivos con un equipo de forma encriptada, y no necesitamos nada muy elaborado. Del lado del servidor, no hay que hacer nada; del lado del cliente, deberemos recompilar el kernel para tener el módulo "shfs.ko" y los utilitarios. Con respecto al parche, no digo nada más. La única aclaración es que la página del proyecto es **http://shfs.sourceforge.net** y que de ahí tendrán que sacar los utilitarios que no están incluidos con el parche sugerido.
La compilación no tiene problemas: como están acostumbrados, pero sin el "**./configure**"; con un mero "**make**" y luego "**make install**" salen andando. Lo único que hace falta instalar es el shfsmount y shfsumount, que ya se imaginarán qué hacen.
Veamos un ejemplo rápido en donde, como usuario root, se utilizan los privilegios de otro usuario, incluyendo su llave pública y privada, para montar un equipo remoto llamado "bubu":

```
# shfsmount -u peperino bubu /mnt/
# df -h lgrep bubu
bubu                20G  870M   18G   5%
 /mnt
# mount l grep bubu
bubu on /mnt type shfs
(version=2,mnt=/mnt,fd=4)
```

Notarán que, según el mount, el espacio disponible en "bubu" es de 18 gigas. Ya les anticipo que no es así. Lamentablemente, es un error que todavía se está tratando de solucionar. Pero el parche no funciona del todo bien por ese lado.
Para desmontar a "bubu" no hay más que hacerlo como si fuera cualquier otro dispositivo. Luego, el sistema trabaja como un floppy o una unidad remota vía nfs. Les dejo algunos ejemplos más para que vean la flexibilidad de **shfsmount**:

Similar al "ssh usuario@host":

```
# shfsmount
peperino@bubu:/opt/prueba /mnt
```

¿El servidor ssh no escucha en el puerto 22?

```
# shfsmount -P 2222
peperino@bubu:/opt/prueba /mnt
```

¿Tienen problemas con la conexión y pierden paquetes?

```
# shfsmount –-persistent
peperino@bubu:/opt/prueba /mnt
```

¿Quieren poder seguir los links simbólicos?

```
# shfsmount –s
peperino@bubu:/opt/prueba /mnt
```

VESA-TNG + BOOTSPLASH

Les voy a ser sincero, veo bien. Me aburro y no me alcanzan las 80x25 líneas de una consola y no arranco las Xs para todo. Lo mejor sería aumentarle la definición a la consola y, por qué no, cambiarle el refresco porque, perdón para la gente que vive en 14'', yo tengo un monitor de 21'' y quiero tener la consola en 1600x1200 en 85 de refresco :=). No me siento más sysop por estar en 80x25 en un monitor de 9''.
El parche que vamos a ver ahora permite mejorar el aspecto visual de nuestra consola. Sin él tendremos ciertas limitaciones cuando trabajemos en ella. Muchas de estas limitaciones están solucionadas cuando instalamos el driver de framebuffer en el kernel y configuramos una definición al inicio del equipo.
Se estarán preguntando: ¿qué es exactamente el driver de framebuffer?
Bueno, se trata de un dispositivo abstracto que representa el frame buffer de la placa de video. Permite que las aplicaciones accedan al hardware de la placa de video a través del mencionado driver. Así, las aplicaciones no tienen que preocuparse por configuraciones.
Para que entiendan un poco más, ¿qué otra aplicación hace algo parecido? Las Xs. El servidor X hace lo mismo: luego de iniciarlo, las aplicaciones ya no se preocupan más por los drivers o por cómo conseguir el diálogo con el hardware; hablan directo con las Xs.
Volviendo al framebuffer, si ya entendieron un poco, termino de aclarar el concepto mediante un ejemplo claro. Con el driver de fb, podremos ver películas utilizando mplayer en consola, ver fotos, tomar screenshots y setear definiciones como 1024x768 en 85 de refresco. ¡Todo en consola sin Xs! Hasta existe un servidor Xs que usa el framebuffer para mostrar las aplicaciones de Xs, que es mucho más liviano que el tradicional.
Como les anticipé, si bien parece mágico, el driver de framebuffer tiene varias limitaciones. Citando lo que dice la página del proyecto, "Vesavesafb-tng es una nueva y más funcional versión del driver de Linux vesafb". Entre otras cosas, permite a los usuarios:

- ★ Setear definiciones con refrescos más altos que los estándar.
- ★ Setear los modos de definición por su nombre, como 1024x768-32@85. Antes había que mirar una tabla entre el modo y el refresco para definirlo al inicio.
- ★ Poder iniciar el soporte en cualquier momento, no necesariamente al inicio.
- ★ Compilarlo como módulo.
- ★ Poder usarlo en la consola y el modo gráfico con los drivers de NVIDIA.

Si llegaron hasta acá, instalar el parche ya es juego de niños. Lo que sí vale la pena remarcar es qué opción hay que elegir en la configuración del kernel para poder comenzar a usar el soporte para vesa-tng.

-Device drivers
-Graphics support
-Support for frame buffer devices
-VESA VGA graphics support

Ahora les queda elegir el método tradicional o el nuevo que proponemos. Opten, obviamente, por el nuevo, y podrán configurar una definición predefinida para la consola con su correspondiente refresco. El formato no es tan difícil de entender en relación con el anterior, en el que había que ver una tabla y era un trastorno memorizar qué significaba cada elemento. Ahora es lógico para cualquiera:

Para 1024x768 en 85 de refresco es 1024x768@85
Para 800x600 en 75 de refresco es 800x600@75

Ninguna ciencia oculta. Otra opción que vale la pena marcar, y que está en el mismo lugar pero al final del menú, es Support for the framebuffer splash. Con ella podrán tener esos "splash screens" que suelen encontrar en distribuciones como SuSE o Mandrake, que inician el equipo con una foto en el fondo (lo mismo que hace el nuevo Gentoo: si elegimos iniciar con soporte para framebuffer, tenemos en las consolas el logo en todo el fondo).
Ahora seguimos como siempre: compilamos el kernel, lo instalamos, generamos una nueva entrada en el grub / lilo y, antes de reiniciar, recordamos sacar las entradas de "vga=" que podamos llegar a tener en el bootmanager. Ya no hace falta establecer la definición mediante esta variable, dado que con el nuevo parche el sistema cambia para mejor; ahora es "video=". Les muestro cómo quedó mi grub teniendo mi viejo kernel con el soporte de framebuffer viejo y su "vga=", y mi nuevo kernel con su valor de "video=":

```
# cat /boot/grub/menu.lst
#
# Sample boot menu configuration file
#

# Boot automatically after 30 secs.
timeout 6

# By default, boot the first entry.
default 0
# For installing GRUB into the hard disk
# title Install GRUB into the hard disk
# root    (hd0,0)
# setup   (hd0)
# Change the colors.
# title Change the colors
# color light-green/brown blink-red/blue
```

Cuando reinicien, van a ver lo linda que les quedó la consola. Ahora ya no parecen tan oscuros y diabólicos, ¿no? ¿Qué les falta? Bueno, antes de llegar a ese punto, veamos cómo jugar un poco con esa consola tan fachera que tenemos ahora.
Usar aplicaciones gráficas en consola no es lo ideal, y no siempre funcionan del todo bien. Habiendo hecho esta aclaración, les muestro algunos ejemplos rápidos para que vean cómo sacarle el jugo al framebuffer, al margen de todo lo ya expuesto:

- ★ Para ver una película en consola con mplayer, tenemos que agregar el switch "-vo fbdev2".
- ★ Para ver imágenes hay una herramienta llamada fbi, que acepta como parámetro una o unas fotos.
- ★ Con fbgrab podremos hacer screenshots de la consola, fbgrab -c 1 archivo.png genera la foto archivo.png de la consola 1.

Para terminar de tener la consola más animada, necesitamos tener esos wallpapers y fuentes TrueType con antialiasing que ya mencionamos que agregan Mandrake, SuSE y Gentoo, entre otros. Para lograrlo, si usamos el grub, nos convendrá releer el ejemplo de configuración que les mostré, donde figura cómo ponerle un "splashscreen" a grub y arrancar bien arriba con una foto en el menú del grub.
Esto que venimos haciendo se llama "bootsplash", y la página del proyecto, misteriosamente, es **www.bootsplash.org**. Pasen por ahí y, luego de bajar los utilitarios y seguir la documentación con el parche ya instado, no tendrán mayores problemas para sacarlo andando solo. No usen el parche que proponen en la página porque no va a funcionar nada; usen sólo los utilitarios.
Si usan Debian, pueden agregar a su **/etc/apt/sources.list** la siguiente línea:

```
deb www.bootsplash.de/files/debian/ unstable main
```

Así podrán tener los utilitarios tan cerca como un "apt-get install". Además, encontrarán muchísimos themes para el bootsplash en ese mismo mirror, no así en la página del proyecto. En freshmeat.net también hay varios themes para bajar.
Para ir terminando, si tuvieron algún problema o algo no funcionó como esperaban y no se animan a escribirme, les dejo la página del proyecto: **http://dev.gentoo.org/~spock/projects/vesafb-tng**.
Seguimos en el próximo número. ★ **Guido Lorenzutti**

UNA ALTERNATIVA A GNU

FreeBSD

ENTRE LOS SISTEMAS OPERATIVOS LIBRES, NUESTRO QUERIDO GNU/LINUX NO ESTA SÓLO. EN ESTA NOTA PRESENTAREMOS BREVEMENTE EL SEGUNDO SISTEMA OPERATIVO LIBRE "UNIX-LIKE" MAS POPULAR: FREEBSD.

Este sistema operativo nació en 1993 como un proyecto cuyo objetivo era producir una "instantánea" usable de BSD. Aunque las primeras versiones tuvieron cierto éxito en el comienzo, en 1994 tuvo que ser reescrito debido a una resolución judicial en la cual se reconocía que el código en el que estaba basado pertenecía a Novell. Así fue como en 1995 vio la luz la primera versión completamente libre: FreeBSD 2.0.2. Hoy en día, FreeBSD es el BSD con más adeptos de la Red, ya que con su sistema de ports (heredado de NetBSD) permite adaptar fácilmente el software diseñado para otros sistemas operativos. Con el sistema de port, la comunidad de FreeBSD vio cómo se incrementaba su base de programas instalables, debido a la gran cantidad de soft para GNU/Linux que se puede "portar" a FreeBSD. La última versión disponible al momento de escribir esta nota es la 5.2.1, en la que me basaré para presentar las explicaciones. Sin más introducciones, pasemos al tema que nos convoca.

EL PROCESO DE INSTALACION

Para instalar FreeBSD necesitamos conseguir la imagen ISO del CD de instalación (es uno solo) de **www.linuxiso.org**, o bien de su página oficial (**www.freebsd.org/es**) en español. Una vez obtenido y grabado, colocamos el CD en alguna unidad y configuramos nuestra PC para que arranque desde ahí. Antes de comenzar con la instalación propiamente dicha, es necesario hacerle espacio a FreeBSD en nuestro disco, para lo cual podemos usar Partition Magic o, mejor aún, QTParted desde un GNU/Linux booteable (como Knoppix, por ejemplo). Les recomiendo que separen al menos 2 gigas de disco para instalar los programas que necesiten. Lo primero que veremos al iniciar el CD será el proceso de carga de kernel, que no difiere mucho de lo que estamos acostumbrados a ver en GNU/Linux. Al cabo de unos minutos, aparecerá un menú desde el cual podremos elegir la instalación que deseemos. Como siempre, lo ideal es utilizar la opción predefinida (presionamos 1 y ENTER), y en caso de que el sistema de instalación no se inicie correctamente, podemos probar a bootear con alguna de las otras.

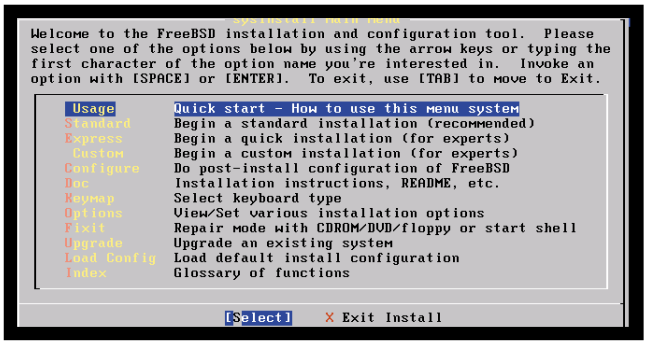
A continuación, pasa una serie de mensajes de kernel, la mayoría sobre detección de hardware, y se inicia el sistema de instalación, que es un menú Ncurses, al mejor estilo instalador de Slackware. Seleccionamos la opción Standard y presionamos ENTER. Luego, se presenta una pantalla que nos informa que estamos a punto de particionar el disco, y nos da dos opciones: presionar la tecla A, en cuyo caso se utilizará **todo el disco**; o presionar la tecla **C** sobre una porción no usada del disco para instalar FreeBSD en ella. Nosotros aplicaremos esta última, ya que es el caso más habitual. Una vez dentro de la herramienta de particionamiento, seleccionamos el espacio libre sobre el que vamos a instalar, presionamos **C** e

```
sio0 port 0x3f8-0x3ff irq 4 on acpi0
sio0: type 16550A
sio1 port 0x2f8-0x2ff irq 3 on acpi0
sio1: type 16550A
fdc0: <Enhanced floppy controller (i82077, NE720E
x3f5 irq 6 drq 2 on acpi0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
orm0: <Option ROMs> at iomem 0xe4000-0xe7fff,0xdc
0000-0xc7fff on isa0
pmtimer0 on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomer
Timecounter "TSC" frequency 1618800768 Hz quality
Timecounters tick every 10.000 msec
md0: Preloaded image </boot/mfsroot> 4423680 byte
acpi_cpu: throttling enabled, 8 steps (100% to 12
ad0: FAILURE - SETFEATURES ENABLE RCACHE status=5
TED>
GEOM: create disk ad0 dp=0xc3144c60
```

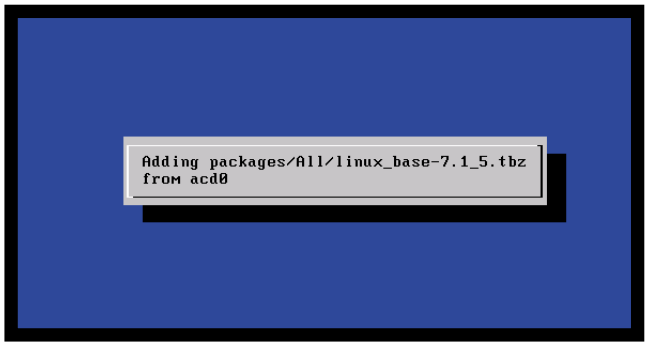
EL PROCESO DE CARGA DEL KERNEL NO DIFIERE MUCHO DE LO QUE ESTAMOS ACOSTUMBRADOS EN GNU/LINUX.

```
Welcome to FreeBSD!

1. Boot FreeBSD [default]
2. Boot FreeBSD with ACPI disabled
3. Boot FreeBSD in Safe Mode
4. Boot FreeBSD in single user mode
5. Boot FreeBSD with verbose logging
6. Escape to loader prompt
7. Reboot

Select option, [Enter] for default
or [Space] to pause timer  7 _
```

AL CABO DE UNOS MINUTOS APARECERÁ UN MENÚ DESDE EL CUAL PODREMOS ELEGIR LA INSTALACION QUE DESEEMOS.



LA HERRAMIENTA DE INSTALACION Y CONFIGURACION, CON SUS MULTIPLES OPCIONES.



UNA VEZ QUE SE TERMINE LA INSTALACION DE PAQUETES, PODEMOS AGREGAR USUARIOS AL SISTEMA.

ingresamos la cantidad de espacio por utilizar. Es necesario aclarar que todo este espacio debe indicarse en este momento, ya que FreeBSD necesita una sola partición (llamada "slice" o porción), dentro de la cual el sistema operativo crea subparticiones para cada punto de montaje. Si deseamos indicar el espacio en MB, debemos ingresar el número seguido por una **M**. Luego se nos pregunta qué tipo de slice crearemos. La opción predefinida es 165 (FreeBSD); presionamos ENTER y la Q para salir y guardar los cambios cuando estemos seguros de que ha quedado como queremos. La siguiente pantalla nos pregunta qué Bootloader queremos instalar. Esta elección depende de la configuración que tengamos en la PC. Si deseamos que se instale BTX (el cargador de arranque de FreeBSD), elegimos la primera opción, que es la que estoy resumiendo. No teman por sus otros SO, ya que FreeBSD los detecta e incorpora como opciones en el BTX.
La pantalla que sigue explica que estamos a punto de crear las particiones BSD dentro del slice generado, y nos da indicaciones con respecto al espacio mínimo que deben tener. Presionamos ENTER, y luego la A, para crear las particiones por defecto, asignándoles el espacio que necesiten. A menos que no deseemos particionar de ese modo, presionamos **Q**.
Ahora ya estamos listos para seleccionar el Distribution Set: los paquetes por instalar. Si disponemos de, al menos, 2 GB de disco, lo ideal es elegir la opción All (todos). En este punto, sysinstall nos informa que se instalará el árbol de ports, que nos permite ampliar nuestro sistema con más de 9.000 programas al costo de 210 megas de disco. Es muy importante instalar los ports, ya que son los que nos permiten instalar soft de manera automatizada mediante una conexión a Internet. Elegimos **All** y presionamos ENTER. Ahora debemos elegir la fuente de instalación, que en nuestro caso será el CD, por lo que seleccionamos la primera opción. Si tenemos una conexión permanente a Internet, y mucho tiempo, podemos elegir la opción FTP, para tener los últimos paquetes disponibles. Es recomendable utilizar el CD y luego actualizar la distro. Después de elegir el medio de instalación, tenemos una última oportunidad de cancelar la instalación. Si elegimos Yes, se procederá a realizar la instalación del sistema con las opciones especificadas. Si nos equivocamos en algo, podemos elegir **No** para volver al menú principal y ejecutar otra vez los pasos que correspondan. Elegimos Yes para continuar.

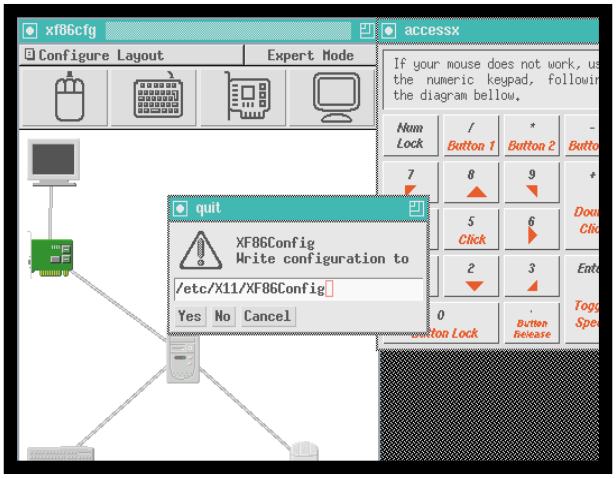
CONFIGURACION GENERAL

Es el momento de ir a comprar nuestra bebida favorita, ya que el proceso puede durar más de 40 minutos. Si observamos lo que sucede, veremos que primero se crean y formatean las particiones elegidas, y luego se instalan los diferentes sistemas seleccionados en la sección anterior. Una aclaración: si por error configuramos algo mal, o luego hay cambios en el sistema que requieran una reconfiguración, siempre podemos volver a entrar en sysinstall escribiendo el comando /usr/sbin/sysinstall como root.

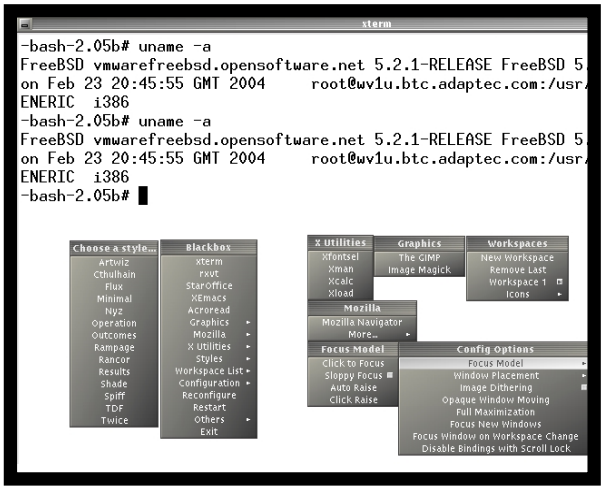
Una vez que se terminan de instalar los paquetes solicitados, el sysinstall nos permite configurar nuestro hardware, así como detalles administrativos del sistema. La primera pregunta que nos hace es si queremos configurar las placas de red o conexiones que tengamos en nuestra PC. Elegimos **Yes**. En el menú que se presenta seleccionamos el tipo de conexiones que deseamos configurar. Las opciones son:

- **lnc0** (puede variar según las marcas de la NIC que tengamos instadas): es la configuración de una placa de red conectada directamente a una LAN.
- **plip0**: sirve para conectarnos a otro sistema FreeBSD usando los viejos cables LapLink (cables Null modem para puerto paralelo).
- **sl0**: para configurar un enlace slip. A menos que tengamos una conexión permanente que use este protocolo, como un router, no nos servirá demasiado.
- **ppp0**: es la opción que debemos elegir si nos conectaremos a ADSL o a un servidor dial-up. Es necesario aclarar que en este punto de la instalación, sólo podemos usar módems dial-up en el puerto serie 1 o 2.

Configuraremos la primera opción, es decir, la placa de red. La primera pregunta es si deseamos que nuestra máquina funcione como Gateway de una red; respondemos No si estamos configurando un cliente. La siguiente pregunta es si deseamos configurar el demonio inetd y los sub-demonios correspondientes. Si deseamos tener algún servidor corriendo en nuestra PC, elegimos Yes; nosotros elegiremos No para continuar con la instalación. La siguiente pantalla nos pregunta si deseamos activar un servidor ssh para loguearnos remotamente. La opción que elijamos dependerá de nuestras necesidades. En general, se activa el servidor



LUEGO DE HACER LAS MODIFICACIONES DEBEREMOS GRABAR EL ARCHIVO DE CONFIGURACION.



UNA VEZ INSTALADO EL SISTEMA OPERATIVO, YA ESTAMOS LISTOS PARA INVESTIGARLO A FONDO.

ssh cuando se piensa trabajar remotamente, o bien cuando la máquina es un servidor sin monitor, por ejemplo, con lo que podremos administrarlo desde alguna terminal remota, sin necesidad de trasladarnos hasta él. Tengan en cuenta también que si elegimos esta opción, nuestra máquina puede ser vulnerable a ataques de crackers que utilicen el servicio SSH.

Luego, **sysinstall** nos pregunta si deseamos permitir accesos FTP anónimos a nuestra computadora. Mi recomendación es decir que **No** y luego configurar el demonio FTP para permitir accesos autenticados si son necesarios. La próxima pregunta es si nuestra máquina será servidora NFS. Excepto que sepamos fehacientemente que necesitamos un servidor NFS en nuestra red, elegimos **No**. Cuando luego pregunta si será un cliente NFS, es una buena idea elegir **Yes**, ya que nunca sabemos a dónde tendremos que conectarnos el día de mañana. Luego se nos pregunta si deseamos personalizar nuestra configuración de consola. Si nos animamos, elegimos **Yes**.

A continuación debemos configurar el reloj de la PC. Elegimos **Yes**, y luego **No** cuando se nos pregunta si deseamos usar UTC. Elegimos América, y la región en la que nos encontremos. Cuando nos proponga una abreviatura para la timezone, elegimos **Yes** para aceptar la propuesta y **No** si queremos usar una personal.

La siguiente pregunta es si deseamos tener compatibilidad con binarios Linux a nivel kernel. Es recomendable activar esta opción si planeamos usar aplicaciones Linux, por lo que elegimos **Yes**. Esto desencadena la instalación del paquete Linux_base y de otros necesarios para tener esta característica.

En el paso siguiente configuramos el mouse en caso de tener uno. Elegimos **Yes**, luego **Enable** para que el sistema lo detecte automáticamente, y lo movemos para verificar que la instalación haya sido exitosa. En caso afirmativo, elegimos **Yes** y luego **Exit** para proceder a la instalación del servidor X.

INSTALACION DEL SERVIDOR X

Si queremos contar con interfaz gráfica, debemos instalar el servidor X eligiendo Yes y, luego, seleccionamos el método de configuración de X que deseamos usar. Sysinstall nos da la opción de instalar paquetes adicionales no incluidos en los seleccionados con anterioridad. Para elegirlos, podemos navegar por el árbol buscando paquetes por categoría, o bien recurrir a la opción All, donde podremos verlos todos ordenados alfabéticamente. Una vez que los elegimos, seleccionamos Install, y el sistema nos mostrará una lista de los paquetes que se instalarán. Elegimos OK si estamos conformes, o Cancel en caso contrario, para volver atrás y agregar o quitar los que consideremos. Ahora es el momento de hacer otro paseo mientras se instalan los paquetes seleccionados.

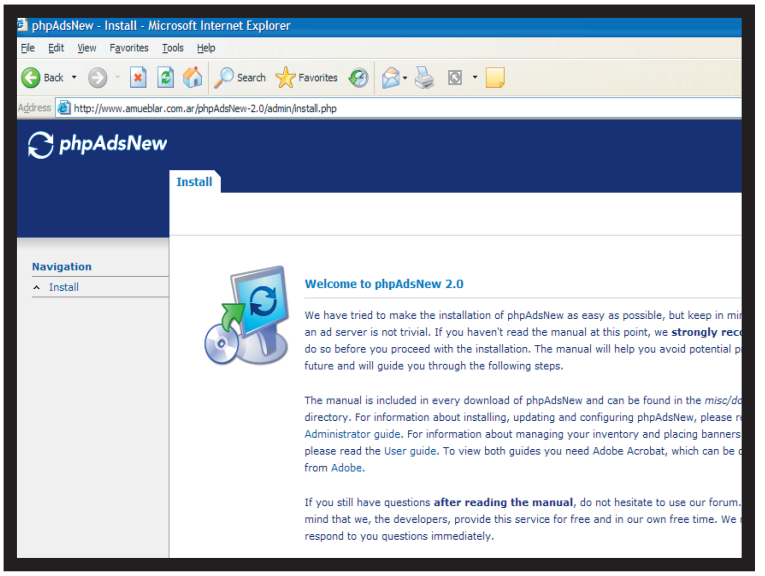
FINALIZANDO...

Una vez que terminó la instalación de paquetes, podemos agregar usuarios a nuestro sistema. Es una buena idea hacerlo para no tener que realizar nuestras tareas diarias con el usuario root. Una vez configuradas las cuentas de usuario, nos pide la contraseña del administrador (root para los vagos); la ingresamos y luego de aceptarla, nos da la opción de ir al menú para configuraciones o cambios de último momento. Ponemos No para arrancar finalmente con el sistema. De todas maneras, siempre podemos volver a ejecutar **sysinstall** mediante el comando **/usr/bin/sysinstall**. Para terminar, elegimos Exit Installer, retiramos el CD de instalación de la lectora y escogemos Yes para que se reinicie el sistema. En la próxima edición de la revista, veremos cómo dar nuestros primeros pasos en este sistema operativo. ★ **Marcos Ibáñez**

HACIENDO PUBLICIDAD EN NUESTRO SITIO

phpAdsNew

La publicidad en Internet difiere en varias formas de la tradicional: por ejemplo, de la que se hace en televisión, que se vende por tiempo de aire; o de la que podemos comprar en un diario, donde el precio varía según el espacio contratado.
En Internet, la publicidad se realiza, principalmente, mediante banners, que en general se venden en función de la cantidad de exposiciones que tienen, es decir, de la cantidad de veces que se los muestra. Esto significa que cuando un potencial cliente desea contratar publicidad en un sitio, contrata cantidad de exposiciones, y no un determinado período de tiempo, ya que no es lo mismo tener un mes de publicidad en un sitio que recibe mil visitas al día, que en uno que recibe cincuenta, ¿verdad?
Otra diferencia importante, además de la interactividad que sólo nos puede ofrecer Internet, es que, a diferencia de otros medios publicitarios, en la Web podemos medir con mayor exactitud el resultado de nuestra publicidad. Por ejemplo, si colocamos publicidad en un programa de televisión, ¿cuánta gente la ve? Realmente, no conocemos con exactitud esa respuesta, ya que sólo podremos estimar la cantidad de personas que ven la televisión en ese momento (usualmente, mediante el rating), pero ese dato nunca será 100% real. Ahora bien, si colocamos un banner en un sitio en Internet, podremos saber exactamente cuántas veces fue mostrado, es decir, cuánta gente lo vio, además de que podremos conocer la cantidad de clics realizados sobre él, entre otros detalles.
Ahora bien, toda esta medición no se hace automáticamente, sino que necesitamos utilizar un programa conocido como Adserver, o servidor de publicidad, que nos permitirá medir las exposiciones y los clics; además, si es realmente bueno, nos dará mucha más información que nos ayudará a segmentar la publicidad con más efectividad y, por consiguiente, nos permitirá lograr mejores resultados (por ejemplo, de nada nos servirá mostrarle publicidad de una radio local a un visitante japonés).



EJECUTANDO EL ASISTENTE DE INSTALACION. ¡QUE COMIENCE LA FUNCION!

UNA SOLUCION DE SOFTWARE LIBRE

El programa que analizaremos en este caso es phpAdsNew, un poderoso Adserver cuya dirección oficial es **www.phpadsnew.com**; por supuesto, está bajo la licencia GPL. Esta herramienta no sólo nos permitirá administrar diferentes banners y mostrarlos en forma rotativa en nuestro sitio web, sino que también nos presentaría una completa estadística sobre cada uno de ellos; además, que posee distintas opciones, como poder filtrar cada banner en forma individual para que se muestre determinados días de la semana, a ciertos horarios, a visitantes con determinada IP o browser, entre otras.
Por último, cabe destacar la funcionalidad de Geotargeting, que nos permite mostrar banners a visitantes de los países o continentes que seleccionemos. Por ejemplo, si en nuestro sitio tenemos un tráfico bastante heterogéneo de distintos países, gracias a esta opción podremos dejar que el sistema se encargue de mostrar los banners en español sólo a usuarios latinos, y aquellos que estén en inglés, a usuarios que hablen esa lengua. También podemos definir cierta segmentación, para que un banner se muestre a visitantes de uno o varios países, por ejemplo, Argentina, México y Chile, y no del resto. Estas son sólo algunas de las características que podrán encontrar en este programa, que es muy completo y sencillo de manejar. Entonces, sin más demora, veamos cómo podemos instalarlo en nuestro servidor.

REQUERIMIENTOS DE INSTALACION

Para la instalación de phpAdsNew no necesitamos demasiados elementos, salvo disponer de php y de una base de datos MySQL, algo que la mayoría de los proveedores de hosting podrá ofrecernos sin problema.

TODO ADMINISTRADOR O WEBMASTER DE UN SITIO WEB TIENE INTERES EN INCLUIR BANNERS DE PUBLICIDAD, YA SEA PARA PROMOCIONAR DIVERSAS PAGINAS DE SU SITIO O PARA GENERAR INGRESOS MEDIANTE LA VENTA DE LOS ESPACIOS PUBLICITARIOS DISPONIBLES, Y ASI, OBTENER LA RENTABILIDAD TAN DESEADA.

PASO 1 ★ OBTENER EL PROGRAMA

Para obtener el programa, ingresamos en **http://phpadsnew.com/two/index.html** y vamos a la sección de download, desde donde elegimos el archivo que queremos bajar: .zip si vamos a descomprimirlo en nuestro equipo local y luego subirlo vía FTP al servidor, o .tar.gz para trabajarlo directamente en el servidor.
Si tenemos acceso ssh en nuestro servidor, podremos obtener el programa directamente utilizando el comando wget en el directorio donde deseamos efectuar la instalación, como vemos a continuación:

```
wget
http://umn.dl.sourceforge.net/sourceforge/phpadsnew/phpAd
New-2.0.tar.gz
```

Paso 2 ★ DESCOMPRIMIR EL PROGRAMA

Luego de obtener el programa, debemos descomprimirlo, ejecutando el comando tar como vemos a continuación:

```
tar xvfz phpAdsNew-2.0.tar.gz
```

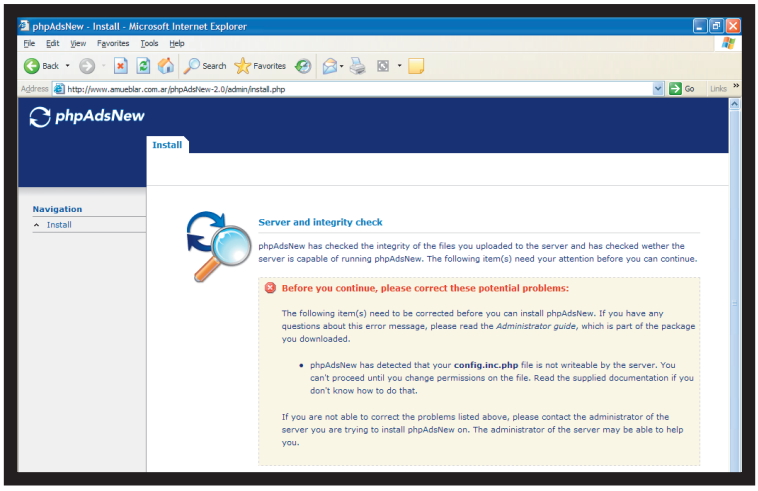
Recuerden que también podemos descomprimir el archivo en nuestro equipo local y subirlo al servidor vía FTP con nuestro programa habitual, si no tenemos acceso SSH.
Luego de haber ejecutado el comando tar, aparecerá el directorio phpAdsNew-2.0, dentro del cual están todos los archivos que conforman el programa.

PASO 3 ★ CREAR LA BASE DE DATOS

Para poder utilizar nuestro programa también debemos crear la base donde residirán los datos correspondientes.
Podremos crearla desde nuestra sesión SSH, o bien desde el panel de control de nuestro proveedor de hosting, por ejemplo, utilizando phpMyAdmin.

PASO 4 ★ EJECUTAR EL ASISTENTE DE INSTALACION

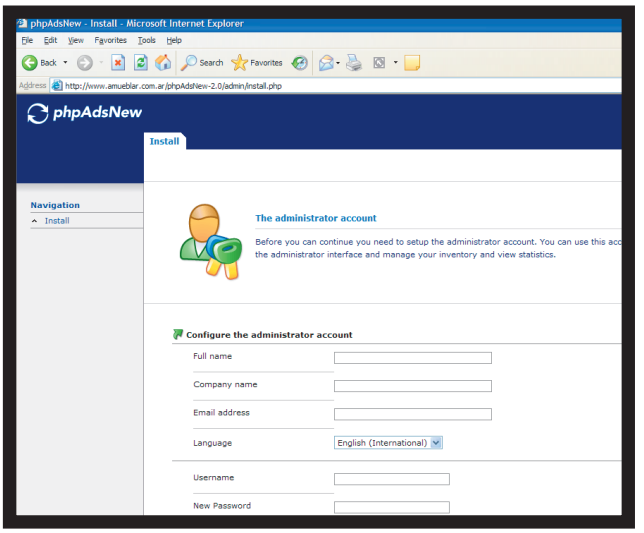
Para ejecutar el asistente de instalación, debemos ingresar en la dirección donde esté instalado el programa, por ejemplo, **www.midominio.com/phpAdsNew-2.0**.
Aparece la primera pantalla del asistente, que nos indica la importancia de leer el manual; la versión para el administrador está disponible en **www.amueblar.com.ar/phpAdsNew-2.0/misc/documentation/administrator-guide.pdf**, en tanto que la versión para el usuario final está en **www.amueblar.com.ar/phpAdsNew-2.0/misc/documentation/user-guide.pdf**.
En este punto, sólo debemos hacer clic en el botón Proceed. Luego aparece la licencia, por lo que volvemos a presionar en Proceed.
Después de estos pasos, puede suceder que se muestre un alerta, ya que el programa detecta que no puede escribir sobre el archivo config.inc.php, que es el archivo donde guardará los datos para la conexión a la base, entre otros. Para solucionar este problema, debemos dar permisos de escritura sobre el archivo config.inc.php, que está residente en el directorio de instalación, mediante el uso del comando chmod (siempre que tengamos acceso SSH), como vemos a continuación:



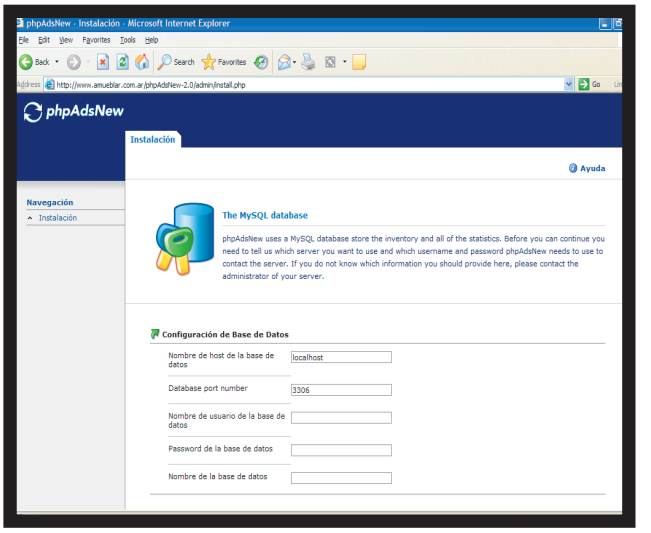
ALERTA: DEBEMOS DAR PERMISOS DE ESCRITURA SOBRE EL ARCHIVO CONFIG.INC.PHP.

```
chmod 777 config.inc.php
```

También podremos cambiar el permiso sobre el archivo si no tenemos acceso SSH usualmente desde nuestro cliente FTP.
Luego de cambiar los permisos, debemos actualizar la página de instalación que estamos viendo, a fin de recargarla. En la pantalla que aparece debemos completar el formulario para configurar nuestra cuenta como administradores. Se nos pedirán los datos básicos, como nombre, e-mail, empresa, nombre de usuario que deseamos para administrar el sistema y contraseña, entre otros.
Ahora, debemos colocar los valores de conexión a nuestra base de datos, como usuario, contraseña y nombre de la base donde residirá la información del programa. Luego hacemos clic en el botón Proceder; si hemos colocado todos los



EN ESTE PASO ESTAMOS CONFIGURANDO LA CUENTA DEL ADMINISTRADOR. PRESTEN ATENCION A LOS DATOS QUE INGRESAN.



CONFIGURANDO LOS VALORES DE CONEXION A NUESTRA BASE DE DATOS.

datos correctamente, aparecerá una pantalla de éxito con el siguiente mensaje:

"Para que phpAdsNew funcione correctamente, deberá asegurarse que el archivo de mantenimiento sea ejecutado cada hora. Para mayor información sobre este tema, lea la documentación".

El archivo de mantenimiento que debemos ejecutar cada hora es el siguiente: **www.dominio.com.ar/directorio_de_instalacion/maintenance/maintenance.php**. Programamos su ejecución utilizando **Cron**, por lo general desde el panel de control de nuestro proveedor de hosting. Un ejemplo sería el siguiente:

```
curl -s -o /dev/null
http://www.dom.com/dir_instalacion/maintenance/maintenance.php
```

Este archivo de mantenimiento automático se encarga de enviar los reportes de estadísticas a los clientes, activar campañas programadas y limpiar los datos estadísticos, entre otras funciones.
Luego debemos hacer clic sobre el botón Proceder para acceder al área de administración, o ingresar directamente desde **www.midominio.com/directorio_de_instalacion**, donde, obviamente, se nos pedirá autentificarnos mediante nuestro usuario y contraseña.

ADMINISTRANDO PHPADSNEW

Luego de la instalación, sólo nos resta comenzar a disfrutar de nuestro programa. Tal vez, al principio, el uso del programa parezca algo más difícil que un simple rotador de banners, pero luego se darán cuenta de que valió la pena haberle dedicado unos minutos más.

Para acceder al área de administración del sistema, debemos ir a la dirección donde hemos instalado el programa, por ejemplo, **www.midominio.com/directorio_de_instalacion**. La administración del programa se divide en cuatro solapas:

★ **Estadísticas**: muestra todo tipo de estadísticas sobre nuestros banners.

★ **Reportes**: permite generar un archivo .csv sobre datos del sistema.

★ **Inventario**: es la parte más utilizada por los administradores, ya que aquí definiremos las zonas de nuestros sitios donde se puede hacer publicidad. También debemos establecer las campañas de nuestros anunciantes, ingresar los banners y demás.

★ **Propiedades**: desde aquí administramos todos los parámetros de nuestro sistema, como la conexión a la base de datos, las opciones de Geotargeting y la personalización de la interfaz, entre otras diversas funcionalidades.

Por último, recuerden mantener su archivo config.inc.php protegido luego de haber definido cómo funcionará el sistema. Además, no olviden el cron del script de mantenimiento.

EN CONCLUSION...

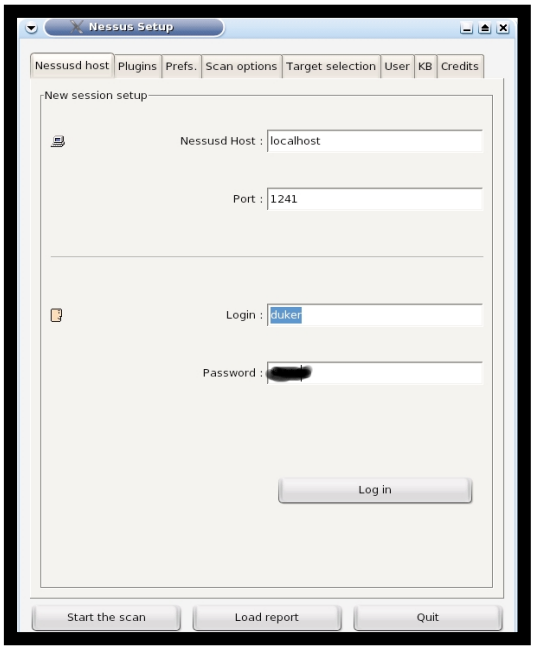
Si tienen algún tipo de inconveniente, pueden consultar la documentación provista por el programa (algunos pdf), que está realmente muy buena (aunque figura en inglés), o pueden recurrir al foro en **forum.phpadsnew.com**. Sin duda, este programa les será de mucha utilidad y podrán obtener todo su potencial para el uso de sus sitios web. Hasta la próxima. ★ **Martín Ramos Monso**

NESSUS SECURITY SCANNER

Auditorías de seguridad

EN ESTA NOTA VEREMOS COMO AUDITAR UN SERVIDOR Y COMO IMPLEMENTAR LAS MEDIDAS NECESARIAS PARA CORREGIR LAS VULNERABILIDADES QUE POSEA. PARA HACERLO, TOMAREMOS COMO BASE UNA COMPUTADORA PENTIUM 1 CORRIENDO MANDRAKE 10 OFFICIAL.

Nessus nació en 1998 para llenar el nicho de una herramienta de seguridad informática que permitiera a la comunidad de software libre realizar auditorías sobre sus sistemas, ya que por ese entonces existía SATAN (*Security Administrator Tool for Analyzing Networks*, herramienta de seguridad para analizar redes), que ya estaba bastante desactualizada.
Hoy en día, Nessus está catalogada como una de las mejores herramienta de su tipo, y hay quienes dicen que es la mejor: nos permite realizar auditorías de las últimas vulnerabilidades descubiertas, ya que sus autores la mantienen en permanente actualización (en la página, una de las desventajas que se señalan al usar instalaciones offline es que nuestra versión no será la última: será la de ayer, ya que se actualiza cada 24 horas).
Al funcionar bajo el paradigma cliente-servidor, es ideal para montar el servidor (nessusd) en nuestro gateway GNU/Linux (o en algún UNIX, el usuario decide) y utilizarlo desde, virtualmente, cualquier plataforma: GNU/Linux, *nix e, incluso, Windows.



ESTA ES LA PANTALLA DE LOGIN DEL CLIENTE NESSUS, DONDE DEBEREMOS INDICAR USUARIO, CONTRASEÑA Y DIRECCION IP DEL SERVIDOR.

MANOS A LA OBRA

Para comenzar con la evaluación, lo primero que debemos hacer es ejecutar el comando **nessusd** desde una consola o bien desde el lanzador de aplicaciones de KDE (o Gnome), presionando **Alt-F2**. Este paso puede obviarse si se lo inicia al arrancar el sistema. Deben tener en cuenta que en la máquina servidor usada para realizar las pruebas (Sempron 2300+, 1 GB RAM, SuSE 9.1), el demonio tardaba al menos dos minutos en arrancar.
Una vez que arranca, iniciamos el cliente ejecutando el comando nessus. Se abrirá un cliente GTK, que nos permite utilizar esta poderosa herramienta con sólo hacer un par de clics (¡por Dios, que windowsero se leyó eso!).
En esta pantalla debemos loguearnos al servidor que acabamos de iniciar, para lo que basta con ingresar el nombre de usuario y la contraseña. Un tip: para poder mover la ventana tomándola desde cualquier parte, hacemos clic en ella mientras mantenemos presionada la tecla ALT. Cuando vean el tamaño de la ventana, entenderán el porqué de esta aclaración.
La primera vez que nos logueemos con el servidor, se presentará una pantalla preguntando cuánto tiempo debe durar el certificado. Elegimos la opción que más nos guste (se recomienda elegir la predeterminada) y, a continuación, hacemos clic en Yes para aceptar el certificado como válido.
Ahora empieza la magia: en la pestaña que se despliega (Plugins), podemos elegir el tipo de ataques a los que someteremos al host objetivo.

Es recomendable que, si vamos a escanear una máquina que nos pertenece (o sobre la que tengamos permiso para realizar la auditoría), elijamos Enable all, que realizará un análisis exhaustivo del equipo. Hay que tener en cuenta que esta opción puede producir un cuelgue del host destino, por lo que sólo debemos usarla en aquellos horarios en los que esa situación no afecte el funcionamiento de la máquina.
En nuestro caso, durante un escaneo con todos los plugins habilitados, uno de los host escaneados empezó a imprimir páginas de pruebas sin parar, ya que una de las vulnerabilidades testeadas por los plugins peligrosos realizaba esa acción.
Si vamos a "investigar" un host que no nos pertenece, tendremos que usar la opción **Enable all but dangerous plugins**, que sólo realizará aquellas pruebas que no arriesguen la estabilidad del sistema.
Además de elegir alguna de las opciones predeterminadas, podemos seleccionar manualmente qué plugins se utilizarán.
Una vez que elegimos las auditorías que se realizarán, hacemos clic en la pestaña **Scan options** y seleccionamos el checkbox que está junto a la palabra **Nmap**. Esto hará que Nessus se valga de **Nmap** para escanear puertos (obviamente, debemos tenerlo instalado), lo que aumenta la eficiencia del escaneo. Con respecto a la pestaña **Prefs**., la dejamos para otra nota más extensa, ya que requiere una explicación muy detallada de las "entrañas" de la pila de protocolos TCP/IP.
El último paso antes de que comience el escaneo propiamente dicho es indicar a Nessus cuál es la URL o dirección IP del host por escanear, en la pestaña Target Selection. Es importante notar que podemos especificar direcciones CIDR, por lo que es fácil escanear una red completa de una sola vez y usando las mismas opciones. Una vez indicado el objetivo, presionamos el botón **Start the Scan**, ubicado en la parte inferior de la pantalla. Para verlo, probablemente tengamos que correr la ventana usando la tecla ALT.

ANALISIS DE LOS RESULTADOS

En las imágenes podemos ver, como ejemplo, el escaneo de toda una subred **192.168.0.0/24**, donde se muestran los siguientes agujeros de seguridad, que pasaremos a analizar:

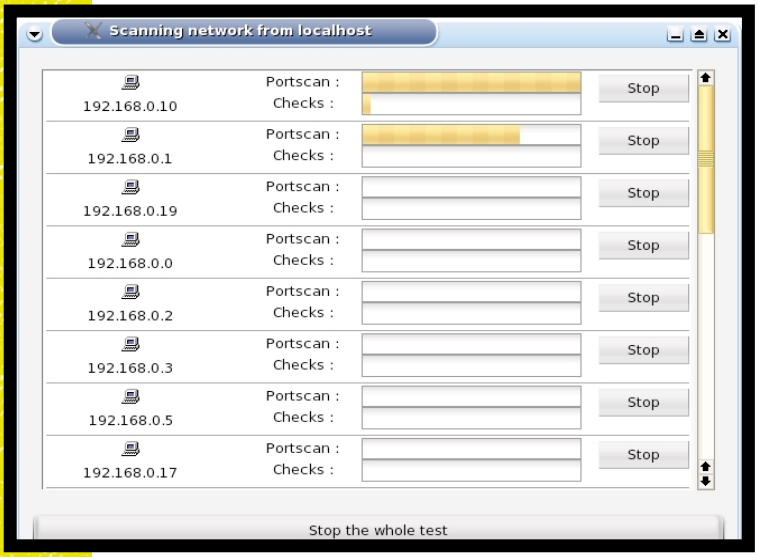
- **Versión de PHP anterior a la 4.3.7**: se nos informa que en versiones de PHP anteriores a la 4.3.7 existen dos bugs que pueden permitir a un atacante remoto la ejecución de código arbitrario en la PC. Para solucionar este problema se sugiere actualizar PHP a la versión 4.3.8.

- **Ssh anterior a la versión 3.7.1**: existe una vulnerabilidad en las versiones de openssh anteriores a la 3.7.1, que, por medio de un buffer overflow, puede permitir a un atacante ejecutar código arbitrario en el host. Solución propuesta: actualizar a ssh versión 3.7.1 al menos.

- **El host está corriendo una versión de PHP Nuke**: el equipo de Nessus recomienda desinstalar este paquete debido a la historia de vulnerabilidades que tiene, y cito: "Aparentemente la seguridad no estaba en la cabeza de la persona que escribió este programa". Incluso se menciona que el autor comenzó una rescritura total del código orientada a la seguridad. Solución propuesta: desinstalarlo y usar otra cosa.
- **Apache anterior a 1.3.29**: existen varias vulnerabilidad que pueden permitir a un atacante ejecutar código arbitrario, mediante mod_rewrite y mod_alias. Solución propuesta: actualizar a la versión 1.3.29.
También se citan, en cada vulnerabilidad, las diferentes listas de seguridad en las que la vulnerabilidad tiene una referencia (ID). Si deseamos aprender mejor cómo funcionan dichas fallas, podemos dirigirnos a los archivos históricos de las listas que se mencionan. En el **Cuadro 2** están las direcciones de las listas más conocidas; una vez que ingresamos en las páginas correspondientes, podremos realizar búsquedas por ID del artículo, y si no nos satisface lo que encontramos, con sólo ingresar el ID completo (CAN-2003-0542, por ejemplo) en Google, hallaremos mucha información sobre ellas.

CARACTERISTICAS DEL SERVIDOR AUDITADO

El servidor auditado es un Pentium 233 MMX (225 reales) con 64 MB de RAM dispuestos en 4 SIMM del 16 MB cada uno. Está equipado, además, con un disco duro Seagate de 10 GB de capacidad, que contiene una partición de Swap de 811 MB y una partición Ext3 de 9,3 GB. Posee dos placas de red Realtek 8139, una de las cuales está conectada a un módem Cisco 677, y la otra, a un concentrador Encore.
El sistema operativo es Mandrake Linux 10 Official, que, entre otros paquetes, tiene instalados MySQL (4.0.15), Apache (1.3.28), PHP-Nuke (7.3), Oscommerce, así como HTB.init para balanceo de carga entre las máquinas de la red.
La función de esta máquina es compartir una conexión a Internet de banda ancha (ADSL 256K) entre los seis equipos que componen la red interna, para lo cual tiene un servidor DHCP, uno DNS y una jerarquía de balanceo de carga basada en colas HTB. A su vez, hostea varias páginas propias de la empresa, las cuales usan el mod_php y MySQL para generar contenidos dinámicos.



EN ESTE EJEMPLO VEMOS COMO ES POSIBLE REVISAR VARIOS HOSTS DE LA RED AL MISMO TIEMPO.

A PARCHEAR EL QUESO GRUYERE

Ahora actualizaremos los paquetes necesarios para securizar el servidor. En mi experiencia, quienes busquen paquetes RPM o DEB deben dirigirse a **www.rpmseek.com**, **www.rpmfind.net** o **www.tuxfinder.com** (a mi parecer, el primero es el mejor). Una ventaja que presenta rpmfind sobre los otros es que podemos realizar la búsquedas por nombre de paquete, y también buscar archivos que estén en los paquetes, lo que muchas veces puede salvarnos la vida cuando el RPM nos informa que falta determinado archivo, pero no sabemos a qué paquete pertenece.

PHP y Apache.

Para actualizar PHP buscamos los paquetes para nuestra distro y los bajamos. Para saber cuáles son los que necesitamos para actualizar nuestro servidor podemos ejecutar:

```
# rpm -qa | grep php (Distribuciones basadas en RPM)
# apt-cache search php | less (Debian y derivados)
```

En el caso de mi servidor de pruebas, el resultado fue:

```
php-mysql-4.3.2-3mdk
php-readline-4.3.2-3mdk
php-calendar-4.3.4-1mdk
libphp_common432-4.3.4-4mdk
php-ini-4.3.4-1mdk
libphp_common432-4.3.3-2mdk
php-gd-4.3.2-4mdk
php-ldap-4.3.2-3mdk
mod_php-4.3.3-2mdk
php-imap-4.3.2-3mdk
```

El paso siguiente consiste en recurrir a algún motor de búsqueda y descargar todos los paquetes solicitados. Es posible que durante la instalación, alguno que otro paquete no se instale debido a una dependencia no resuelta, en cuyo caso deberemos buscar las dependencias e instalarlas. Una vez que tenemos las versiones actualizadas de los paquetes procedemos a instalarlas.

En el caso del servidor de pruebas utilizado, el orden correcto es el siguiente:

```
# rpm -Uvh php-ini-4.3.9-2mdk.noarch.rpm
# rpm -Uvh libphp_common432-4.3.9-2mdk.i586.rpm
```

Aquí se produjo un error indicando que faltaba la librería Insternacionalizacion, versión 3, que se instala con el comando:

```
# rpm -Uvh libintl3-0.14.1-9mdk.i586.rpm
```

Volvemos a intentar el libphp_common, y como se instala, seguimos con los demás paquetes.

```
# rpm -Uvh mod_php-4.3.9-1mdk.i586.rpm
```

Este paquete nos indica que deberemos instalar los paquetes apache, apache-common y libgdbm3 para continuar.

```
# rpm -Uvh libgdbm3-1.8.3-2mdk.i586.rpm
# rpm -Uvh –nodeps apache-common-1.3.31-1mdk.i586.rpm
```

(Tuve que usar este paquete porque no pude encontrar el apache-common-1.3.33 necesario). Luego vemos que apache-common necesita GLIBC 2.3.4, que no pude encontrar, pero luego de probar varias configuraciones, funcionó sin problemas con GLIBC 2.3.3.

```
# rpm -Uvh glibc-2.3.3-22mdk.i586.rpm
```

Finalmente, instalamos el resto de los paquetes:

```
# rpm -Uvh –nodeps apache-common-1.3.31-1mdk.i586.rpm
# rpm -Uvh –nodeps apache-modules-1.3.33-2mdk.i586.rpm
# rpm -Uvh –nodeps apache-1.3.33-2mdk.i586.rpm
# rpm -Uvh php-ldap-4.3.9-1mdk.i586.rpm
# rpm -Uvh php-mysql-4.3.9-1mdk.i586.rpm
# rpm -Uvh php-readline-4.3.9-1mdk.i586.rpm
# rpm -Uvh php-calendar-4.3.8-1mdk.i586.rpm
```

Para instalar los dos paquetes restantes (php-ldap y php-gd), debemos instalar previamente libgd2-2.0.27-3mdk.i586.rpm y libc-client-php0-2004-1mdk.i586.rpm, lo hacemos de la manera tradicional:

```
# rpm -Uvh libgd2-2.0.27-3mdk.i586.rpm
# rpm -Uvh libc-client-php0-2004-1mdk.i586.rpm
# rpm -Uvh php-gd-4.3.9-1mdk.i586.rpm
# rpm -Uvh php-imap-4.3.9-1mdk.i586.rpm
```

Una vez hecho todo esto, reiniciamos el servidor mediante el comando service httpd restart, y listo. Como habrán observado, aunque no es muy difícil realizar el proceso, es bastante tedioso.

OpenSSH

La instalación de un servidor de ssh más nuevo es similar al procedimiento anterior. Primero buscamos los paquetes que tengan que ver con ssh:

```
# rpm -qa | grep ssh (RPM)
# apt-cache search ssh | less (DEB)
```

En el servidor la respuesta fue:

```
openssh-3.6.1p2-12mdk
openssh-server-3.6.1p2-12mdk
openssh-clients-3.6.1p2-12mdk
```

Una vez obtenidas las actualizaciones correspondientes, procedemos a instalarlas. Desde una shell, siendo root, ingresamos las siguientes líneas:

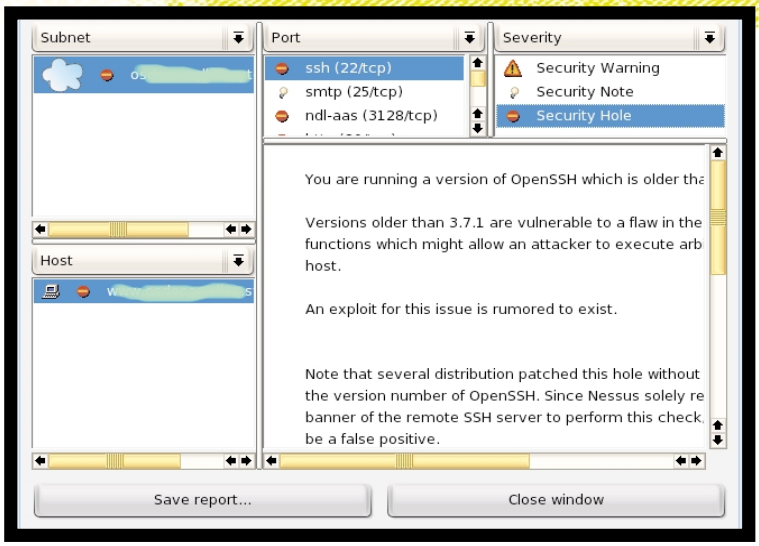
```
# rpm -Uvh —nodeps openssh-3.9p1-3mdk.i586.rpm
# rpm -Uvh openssh-server-3.9p1-3mdk.i586.rpm
# rpm -Uvh openssh-clients-3.9p1-3mdk.i586.rpm
```

Si observamos detenidamente el primer comando, notaremos que hay que decirle al rpm que no controle las dependencias del paquete openssh, ya que el que está instalado es necesario para las versiones instaladas de openssh-clients y openssh-servers, por lo que no nos deja instalarlo. Una vez hechos todos estos pasos, basta con reiniciar el servidor ssh con service sshd restart, y nuestro sistema no tendrá más agujeros de seguridad. El último paso consiste en volver a escanear el servidor para asegurarnos de que han desaparecido los agujeros de seguridad. Si observan la **Figura 4**, notarán que no sólo han desaparecido los agujeros de seguridad, sino que también lo han hecho las advertencias (warnings), y únicamente quedan notas que dicen que el servidor tiene abiertos esos puertos, y hay servidores escuchando por ellos.

EN RESUMEN...

Para terminar, me gustaría revisar los pasos necesarios para securizar un servidor:

- Escanearlo, en lo posible fuera de horarios críticos, con la opción Enable All, de Nessus.
- Buscar en Internet la solución propuesta para los agujeros de seguridad que se encuentren; generalmente, consiste en instalar un par de RPMs.



LUEGO DE LA AUDITORIA, NESSUS NOS INFORMA LAS VULNERABILIDADES DEL SISTEMA Y NOS ACONSEJA SOLUCIONES.

- Bajar e instalar las últimas versiones de los paquetes vulnerables e instalarlos, y hacer lo mismo con las dependencias necesarias.
- Volver a escanear la máquina con Nessus. Es fundamental realizar este paso para asegurarnos de que los cambios se han llevado a cabo exitosamente.

Espero que este artículo les haya sido útil, y les dejo la dirección de una lista de correo del Grupo de Usuarios Linux de la Facultad Regional Resistencia, de la UTN (Argentina), para toda persona interesada en el mundo de GNU/Linux y el software libre: www.freelists.ors/list/gulfrre. No duden en suscribirse y postear cualquier tipo de inquietudes, que les serán respondidas lo antes posible.
Para terminar, les recuerdo que un sistema 100% seguro no existe, pero si siguen los pasos detallados en estas páginas, el suyo puede llegar al preciado 99%. Hasta la próxima. Be Free. ★ **Marcos Gabriel Ibáñez**

TABLA 1 ★ SITIOS DE CONSULTA SOBRE VULNERABILIDADES

URL	DESCRIPCION
www.cve.mitre.org	Comon Vulnrabilities and Exposures
www.securityfocus.com/bugtraq	Lista de correo de Bugtraq
xforce.iss.net/xforce/xfdb	Base de datos de vulnerabilidades de Internet Security Systems
www.redhat.com/archives/redhat-watch-list	Lista de vulnerabilidades de RedHat
lists.debian.org/debian-security announce	Vulnerabilidades de Debian
www.suse.de/de/security	Vulnerabilidades de SuSE
www.mandrakesecure.net	Vulnerabilidades de Mandrake
www.turbolinux.com/security	Vulnerabilidades de Turbo Linux
lists.trustix.org/pipermail/tsl-announce	Vulnerabilidades de Trustix
www.osvdb.org	Comunidad Open Source que brinda información técnica de todo tipo

PROGRAMANDO EN C

Un servidor de chat

La idea general es tener un servidor corriendo en un puerto X (configurable) que acepte conexiones y permita a los usuarios escribir y repetir todo lo que escriben a los demás usuarios.
Antes de nada, definiremos que los usuarios se conectarán a nuestro servidor vía Telnet, es decir que desde sus conexiones escribirán cadenas de texto, y esto es lo que nos llegará al servidor. Así evitaremos escribir un cliente para nuestro sistema.
Luego definiremos el protocolo. El usuario deberá escribir una barra (/) seguida del comando que quiera ejecutar y, luego, dejando un espacio, los parámetros de ese comando. Definiremos también que nuestro sistema tendrá tres comandos: **msg**, **nick** y **exit**. El primero se utilizará para mandar mensajes (por ejemplo: **/msg** hola a todos); el segundo, para cambiarse a sí mismo el nombre en el sistema (**/nick soy_yo**) y, finalmente, exit, para desloguearse (**/exit**).
Por practicidad, permitiremos que el usuario omita el comando y, por defecto, utilice el comando msg.
El sistema es lo suficientemente flexible como para que sea sencillo agregar comandos.

EL ALIMENTO DE LA BESTIA

Antes de empezar a analizar el servidor en sí, veremos la estructura de nuestros datos. Para este sencillo servidor sólo necesitaremos tener una lista con todos los clientes conectados. La estructura de cada nodo de nuestra lista será (archivo **nodo_cliente.h**):

```
#ifndef __NODO_CLIENTE_H__
#include "socket_buffer.h"
typedef struct {
        char *nick;
        SocketBuffer *socketBuffer;
} stTNodoCliente;

#endif
```

De cada cliente tendremos, entonces, el "nick", con el cual se conecta, y otra estructura llamada **SocketBuffer**, que está definida como (del archivo socket_buffer.h):

```
typedef struct {
        int socket;
        char *buffer;
        unsigned int tamanioBuffer;
        unsigned int bytesLeidos;
} SocketBuffer;
```

Como podemos observar, en esta estructura guardaremos el socket propiamente dicho y otras tres variables para administrar un buffer (**buffer** tiene los datos, **tamanioBuffer** nos dice cuánto ocupa, y **bytesLeidos** nos indica hasta dónde está ocupado). La idea de utilizar un buffer entre la entrada directa del socket y nuestro programa se debe a que nadie nos asegura que, cuando leemos, hemos leído todo lo que nuestro programa espera como "un comando". El cliente del usuario puede o no haber enviado la línea completa, o quizá no nos llegue todo de una sola vez. De esta forma, hasta que no se complete la línea, iremos acumulando en el buffer toda la entrada.

EMPEZAR POR EL PRINCIPIO

Comenzaremos por ver el archivo **main.c**, en el cual se encuentra al estructura general del sistema:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <sys/socket.h>
#include "listaClientes.h"
#include "comandos.h"

short validarArgumentos(int argc, char
**argv);
void atraparSeniales();
void marcarSalida(int senial);
int crearServidor(short puerto);
int ejecutarSelect(int socketServidor,
ListaClientes *listaClientes, fd_set
*setLectura);
void agregarNuevaConexion(int socket
Servidor, ListaClientes *listaClientes);
int salir = 0;

int main(int argc, char *argv[]) {
        ListaClientes listaClientes;
        NodoListaClientes *iterador
Clientes;
        stTNodoCliente *nodoCliente;
        fd_set setLectura;
        int socketServidor, error;
        short puerto;

        puerto = validarArgumentos
(argc, argv);
        socketServidor = crearServidor
(puerto);
        atraparSeniales();
        inicializarListaClientes
(&listaClientes);

        while(! salir) {
                error =
ejecutar
Select(socketServidor, &lista
Clientes,
&setLectura);
                if(error ==
-1) {

salir = 1;
```

EN ESTA OPORTUNIDAD VEREMOS UN EJEMPLO PRACTICO DE PROGRAMACION EN C: DESARROLLAREMOS UN PEQUEÑO SERVIDOR DE CHAT. ESTE DEBERA PERMITIR LA CONEXIÓN DE CUALQUIER CANTIDAD DE CLIENTES, VIA TELNET, QUE SE COMUNICARAN ENTRE ELLOS EN UNA UNICA SALA VIRTUAL. ADEMAS, CREAREMOS UN SISTEMA PARA PERMITIR QUE EL USUARIO INGRESE COMANDOS BASICOS, DE MANERA QUE SEA FACILMENTE EXTENSIBLE.

```
                        continue;
                        }
                        if(FD_ISSET
(socketServidor, &setLectura)) {
                        agregar
Nueva
Conexion(socketServidor, &lista
Clientes);
                        }
                        iterador
Clientes =
darIterador(&listaClientes);
                        while
(iteradorClientes)
{
                        nodoCliente
= darActual(iteradorClientes);
                        siguiente
(&iteradorClientes);

if(FD_IS
SET(nodoCliente->socketBuffer
>socket,
&setLectura)) {

procesarEntrada(nodoCliente,
&listaClientes);
                                }
                        }
            }
            return 0;
}

short validarArgumentos(int argc,
char **argv) {
            if(argc != 2) {
                        printf(
sintaxis: %s
<puerto>\n", argv[0]);
                        exit(1);
            }
            return atoi(argv[1]);
}

void marcarSalida(int senial) {
            salir = 1;
}

void atraparSeniales() {
            if(signal(SIGINT, marcar
Salida
)==SIG_ERR) {
                        perror
("[ERROR
signal]");
                        exit(3);
            }

            if(signal(SIGUSR1,
marcarSalida
)==SIG_ERR) {
                        perror
("[ERROR
signal]");
                        exit(3);
            }
}

int crearServidor(short puerto) {
            int iOptname = 1;
            int socketServidor;
            struct sockaddr_in
addr_serv;
            addr_serv.sin_family =
AF_INET;
            addr_serv.sin_addr
.s_addr =
htons(INADDR_ANY);
            addr_serv.sin_port =
htons
(puerto);
            bzero(&(addr_serv.
sin_zero),8);

            if((socketServidor =
socket(AF_I
NET, SOCK_STREAM, 0)) == -1) {
                        perror
("[ERROR
socket]");
                        exit(2);
            }

if(setsockopt(socketServidor,
SOL_SOCKET, SO_REUSEADDR,
&iOptname, sizeof(iOptname))
== -1) {
                        perror
("[ERROR setsoc
kopt]");
                        exit(2);
            }

            if(bind(socketServidor,
(struct sockaddr
*) &addr_serv, sizeof(addr_serv))
== -1) {
                        perror("[ERROR bind]");
                        exit(2);
```

```
        }

        if(listen(socketServidor, 5) == -1) {
                perror("[ERROR bind]");
                exit(2);
        }
        return socketServidor;
}

int ejecutarSelect(int socketServidor,
ListaClientes
*listaClientes, fd_set *setLectura) {
        int max = socketServidor;
        struct timeval stTimeout;
        NodoListaClientes *iterador
Clientes =
darIterador(listaClientes);
        stTNodoCliente *nodoCliente;

        FD_ZERO(setLectura);
        FD_SET(socketServidor,
setLectura);
        while(iteradorClientes) {
                nodoCliente =
darActual
(iteradorClientes);
                FD_SET(nodoCliente->
socketBuffer->socket, setLectura);
                if(nodoCliente->socket
Buffer->socket > max)
                max = nodo
Cliente->socketBuffer->socket;
                siguiente(&iterador
Clientes);
        }

        stTimeout.tv_sec = 3;
        stTimeout.tv_usec = 0;
        return select(max + 1, setLectura,
NULL, NULL, &stTimeout);
}

void agregarNuevaConexion(int socket
Servidor,
ListaClientes *listaClientes) {
        stTNodoCliente nuevoCliente;
        struct sockaddr_in stSockAddr;
        int iLong = sizeof(struct
sockaddr_in), socket;
        socket = accept(socketServidor,
(struct sockaddr *) &stSockAddr, &iLong);
        nuevoCliente.socketBuffer =
crearSocketBuffer(socket, 4096);
        nuevoCliente.nick = (char *)
malloc (20);
        sprintf(nuevoCliente.nick,
"Guest%d", socket);
        agregarNodoListaClientes
(listaClientes, &nuevoCliente);
        printf("El usuario %s ingreso
al chat\n", nuevoCliente.nick);
}
```

Observemos primero la función **main**. Las primeras cuatro sentencias (salteando la definición de variables) inicializan el servidor.
La función validarArgumentos se encargará de verificar que se haya pasado por parámetro el puerto sobre el cual se quiere que funcione nuestro servidor. En caso de que todo esté OK, devolverá el valor del puerto.
El siguiente paso es crear un socket (**crearServidor**) desde el cual podamos aceptar nuevas conexiones.
Luego atraparemos (**atraparSeniales**) las señales **SIGINT** y **SIGUSR1** para que, al recibirlas, terminen la ejecución del servidor.
Finalmente, inicializaremos nuestra lista de clientes conectados para poder utilizarla.
El ciclo principal se ejecutará mientras la variable salir sea 0. Esta variable se modifica cuando se atrapa alguna de las señales mencionadas antes.
Dado que estaremos atendiendo una cantidad indeterminada de clientes, utilizaremos un servidor multiplexado, mediante la función select. Esto significa que esta función estará leyendo de todas las conexiones y nos avisará en cuáles hay algo para leer (ver apartado sobre la función ejecutarSelect). La función **ejecutarSelect** registrará en la variable setLectura todas las conexiones que tengan algo en la entrada. Para comprobar si hay algo para leer, utilizaremos la macro **FD_ISSET**. El primer if revisará contra el socket servidor (desde aquí se aceptan nuevas conexiones) y agregará una nueva conexión a nuestra lista de conexiones (ver apartado de **agregarNuevaConexion**). Luego recorrerá todas las conexiones existentes para ver cuáles han enviado datos, y en ésas llamaremos a la función procesarEntrada para que ejecute el comando pedido.

MULTIPLEXAR EL SERVIDOR

La frase "multiplexar un servidor" se refiere a la posibilidad de escuchar a varios clientes simultáneamente utilizando un mismo camino de ejecución. De esta forma, teniendo un solo proceso e hilo, podremos atender a N clientes. Para lograrlo, Linux nos provee de una función llamada select, a la cual le diremos sobre qué archivos (un socket es, en definitiva, un archivo) queremos escuchar, y se bloqueará hasta que haya algo para leer en cualquiera de ellos. Mientras el programa se bloquea, no consume ningún recurso.
En nuestro programa encontraremos la llamada a select dentro de la función ejecutarSelect. En la última línea veremos que esta función tiene cinco parámetros. Los parámetros dos, tres y cuatro son los set de archivos por escuchar, para lectura, escritura y error, respectivamente. Si no deseamos que select espere por alguno de estos eventos, dejamos el parámetro en **NULL**.
El primer parámetro de **select** es el mayor archivo (recuerden que los archivos o socket se representan con un int) más uno.
El último parámetro determina un tiempo máximo de espera. Si, agotado este tiempo, no se ha encontrado nada para leer, select saldrá forzosamente de la espera. Esta característica es útil porque permite que nuestro servidor haga tareas que se ejecuten periódicamente.

Para manejar los set utilizaremos una variable de tipo **fd_set** y las macro **FD_ZERO** y **FD_SET**. Según el manual de select (man select), es necesario llenar el set con todos los descriptores para escuchar antes de cada llamada; por lo tanto, esto debe hacerse siempre. Para armar un set empezaremos por dejarlo vacío mediante la macro **FD_ZERO**(set). Luego, para cada socket que queramos agregar, llamaremos a la macro **FD_SET** pasando el socket y el set. En nuestra función agregaremos el socket del servidor y luego todos los sockets de los clientes. Cada vez que recorramos un nodo de la lista, verificaremos si éste es el mayor, ya que esta información es necesaria.

CREAR EL SOCKET SERVIDOR

La función crearServidor será la encargada de levantar el servidor. Para "crear" un servidor hace falta crear un socket que esté enlazado y escuchando a un puerto. Este socket se utilizará solamente para aceptar conexiones (cuando se acepta una conexión, se crea un nuevo socket por el cual se puede leer y escribir desde y hacia el cliente). Para crear un nuevo socket (tanto para cliente como para servidor) se utiliza la función socket, que tiene tres parámetros. El primero especifica el dominio; puede ser un socket que funcione sobre Internet con protocolo IPv4 (que es el que utilizaremos, **AF_INET**) o sockets locales (**PF_UNIX**). Luego se debe especificar el tipo de socket. En nuestro caso utilizaremos **SOCKET_STREAM**, que nos provee de una forma segura de conexión bidireccional. Esto quiere decir que el sistema nos asegura que la información llegará en el mismo orden en que fue enviada, y nosotros sólo debemos preocuparnos por leer y escribir. Finalmente, podemos especificar el protocolo que deseamos usar; por lo general, existe un único protocolo según el dominio y el tipo, de modo que, al darle valor 0, automáticamente se elegirá el protocolo correspondiente.

La siguiente función será bind. Omitiremos por un momento la función del medio, que relacionará el puerto que le especifiquemos con el socket. De esta forma, el núcleo Linux sabrá que ese puerto está ocupado y que sólo será posible acceder a él a través de ese socket. Para indicar el puerto es necesario crear una estructura de tipo **sockaddr_in**. En ella deberemos cargar sobre qué dominio estamos trabajando (**AF_INET** para internet) en el atributo sin_family; el número del puerto en sin_port, escrito en orden de red (usamos la función htons, host to network short, para cambiar el orden de los bits de la variable de tipo short al orden de bits de la red) y, finalmente, el atributo sin_addr.s_addr, que nos permite especificar desde qué IPs aceptaremos conexiones. Lo más normal es utilizar la constante INADDR_ANY para aceptar desde cualquier lado. La estructura sockaddr_in debe completarse con 8 bytes de ceros (0) en el atributo sin_zero, de modo que concuerde con la estructura original sockaddr. Es por eso que, en la llamada a bind, es necesario hacer un cast a dicho tipo. Esto es bastante feo pero fue necesario para mantener la compatibilidad hacia atrás.
Luego del bind, sólo falta hacer una llamada a listen para que el socket esté listo para aceptar conexiones. La llamada a listen únicamente necesita el socket (antes se le debe haber hecho un bind al socket en cuestión) y la cantidad de conexiones máximas que se tendrán en la cola antes de que sean aceptadas (si llegan más, serán rechazadas).
Entre la llamada a socket y bind hemos utilizado la función setsockopt, para setear una opción sobre el socket. Esta función toma varios parámetros: el socket, sobre qué nivel se quiere aplicar (**SOL_SOCKET** para que se aplique a nivel del socket; también puede ser a nivel de protocolo), la opción que se quiere activar, un entero con un valor 1 para activarla (0 para desactivar) y, por último, el tamaño de la variable utilizada (ya que la interfaz dice que se debe pasar un void). La opción que activamos aquí es **SO_REUSEADDR**, la cual especifica que si el servidor se muere sin cerrar el socket que escucha al puerto vinculado, este puerto podrá ser utilizado inmediatamente por cualquier otro programa que lo requiera. Esta opción es útil cuando se está depurando el servidor y éste explota (clásico Segmentation Fault). De esta forma, podremos levantar nuevamente el servidor sin necesidad de esperas. Sin esta opción, el puerto quedará vinculado a un socket inexistente, y Linux necesitará un tiempo indeterminado (sin poder forzarlo) para darse cuenta de que ese socket no está en uso, y entonces liberarlo.

AGREGAR NUEVOS CLIENTES

Cuando un cliente se conecte, encontraremos que en **socketServidor** hay algo para leer, y se disparará la función **agregarNuevaConexion**.
Para crear el socket de la nueva conexión utilizaremos la función accept, que toma como primer parámetro un socket servidor (o sea **socket + bind + listen**), un apuntador a una estructura de tipo sockaddr y el tamaño de dicha estructura. Esta función nos devolverá un nuevo socket desde el cual podemos leer y escribir con este cliente. La estructura **sockaddr** contendrá información del cliente y no es necesaria para la comunicación.
Luego de haber aceptado la conexión, completamos la información del nodo cliente que agregaremos a la lista. Para hacerlo, inicializaremos nuestro socketBuffer mediante la función crearSocketBuffer, la cual guardará el socket e inicializará el buffer con el tamaño que le pasemos.
A continuación, en el campo nick, le pondrá un nick por defecto, que luego el cliente podrá cambiar.
Una vez que el nodo está completo, sólo resta agregarlo a la lista de clientes y ya estará disponible para comunicarse con el resto de los usuarios.

SISTEMA DE COMANDOS

Según el protocolo que hemos definido, todos los mensajes del cliente tendrán la forma: **/comando argumentos**. Por lo tanto, buscaremos una manera genérica de definir un comando, para que sea sencillo agregar otros. Para hacerlo, definiremos una estructura que nos permita representar un comando cualquiera (del comando.h):

```
typedef struct {
        char *nombre;
        void (*comando)(char*, stTNodoCliente*,
ListaClientes*);
} Comando;
```

Esta estructura tiene dos atributos: el primero es el nombre del comando (por ejemplo, exit para salir) y el segundo es un apuntador a una función. Esta función recibirá como parámetros los argumentos enviados por el cliente, el nodo del cliente que lo envió y la lista completa de clientes. De este modo, todos los comandos tendrán la información necesaria para funcionar.

Luego, en el archivo **comando.c** definiremos un arreglo con todos los comandos existentes:

```
Comando comandos[] = {
        {"exit", comandoDesconectar},
        {"msg", comandoMensaje},
        {"nick", comandoCambiarNick}
};
```

Así, para encontrar cualquier comando, sólo hay que buscar en el arreglo por el atributo nombre, y ya tendremos un apuntador a la función que se debe ejecutar.

INTERPRETANDO EL PROTOCOLO

La función que interpreta el protocolo y ejecuta el comando es procesarEntrada, que está definida en el archivo comando.c que se muestra a continuación:

```
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
```

```
#include “listaClientes.h”
#include “comandos.h”
Comando comandos[] = {
        {“exit”, comando
Desconectar},
        {“msg”, comando
Mensaje},
        {“nick”, comando
CambiarNick}
};

#define CANT_COMANDOS (sizeof
(comandos) / sizeof(Comando))


void procesarEntrada(stTNodo
Cliente *cliente, ListaClientes
*listaClientes) {
        Comando *ejecutar;
        char *ingreso,
resultado, *comando, *args;
        ingreso = cpLeer
SocketBuffer(cliente->socket
Buffer, &resultado);
        switch(resultado) {
        case LEER_COMPLETO:
                if(ingreso[0]
== ‘/’) {

ingreso++;
        args = ingreso + strcspn(ingreso,
“ \0”);

comando = (char *) malloc (1
+ (args - ingreso));

memset(comando, 0, 1 + (args -
ingreso));

strncpy(comando, ingreso,
args - ingreso);

args++;

ingreso—;
                } else {

args = ingreso;

comando = strdup(“msg”);
                }
                ejecutar =
(Comando *) bsearch(comando,
comandos, CANT_COMANDOS,
sizeof(Comando), comparar
Comandos);
                if(NULL !=
ejecutar) {

ejecutar->comando(args, cliente,
listaClientes);
                } else {

enviarMensaje(cliente, “server”,
“Comando desconocido”);
                }

                free
(ingreso);
                free
(comando);
                break;
        case LEER_
SOCKETCERRADO:
        case LEER_BUFFERLLENO:
                borrarNodoListaClientes
(listaClientes, cliente);
        }
}

int compararComandos(const void
*cmd1, const void *cmd2) {
        char *nombre =
(char *) cmd1;
        Comando *comando2
= (Comando *) cmd2;
        return strcmp(nombre,
comando2->nombre);
}

void comandoCambiarNick(char
*parametros, stTNodoCliente
*cliente,   ListaClientes *lista) {
        free(cliente->nick);
        cliente->nick = strdup
(parametros);
}

void comandoDesconectar(char
*parametros, stTNodoCliente
*cliente,   ListaClientes *lista) {
        borrarNodoLista
Clientes(lista, cliente);
}

void comandoMensaje(char
*parametros, stTNodoCliente
*cliente,   ListaClientes
*listaClientes) {
        NodoListaClientes *ite-
radorClientes =
darIterador(listaClientes);
        stTNodoCliente *nodo-
Cliente;
        while(iteradorClientes) {
        nodoCliente = darActua-
l(iterador
Clientes);
        enviarMensaje(nodo-
Cliente,
cliente->nick, parametros);
        siguiente(&iterador-
Clientes);
        }
}

void enviarMensaje(stTNodoCliente
*nodoCliente, char *nick,
char *mensaje) {
                write(nodo
Cliente->socketBuffer->
socket, nick, strlen(nick));
                write(nodo
Cliente->socketBuffer->
socket, “> “, 2);
                write(nodo
Cliente->socketBuffer->
socket, mensaje, strlen
(mensaje)+1);
                write(nodo
Cliente->socketBuffer->
socket, “\r\n”, 2);
}
```

La función procesarEntrada comienza leyendo desde el socket mediante **cpLeerSocketBuffer**. Esta función, como veremos luego, nos devolverá una cadena con toda la línea que envió el usuario. Sin embargo, no siempre regresará lo que ingresó el usuario; en realidad, el resultado depende de

lo que informe el segundo parámetro. Si éste devuelve la constante **LEER_COMPLETO**, significa que la línea está disponible. Otra posibilidad es que devuelva **LEER_INCOMPLETO**, lo que significa que se leyó algo, pero la sentencia no está completa. Si devuelve **LEER_BUFFERLLENO**, quiere decir que el usuario envió una línea mucho más grande de lo que admite nuestro buffer. Finalmente, si dice **LEER_SOCKETCERRADO**, es que se cortó la conexión con el cliente.
En los últimos dos casos eliminaremos al cliente de nuestra lista de conexiones. Analicemos el caso más complejo, cuando poseemos una entrada completa.
La primera parte consiste en dejar en la variable comando la cadena con el comando ingresado, y en la variable args, los argumentos.
Para esto, verificaremos si la cadena comienza con una barra (/). Cabe recordar que permitiremos a los usuarios enviar mensajes sin obligarlos a usar el comando **/msg**. Si no está la barra, directamente todo lo ingresado serán los argumentos y, por defecto, pondremos a comando con la cadena "msg". En la otra posibilidad, cortaremos la cadena hasta el espacio y copiaremos esa primera parte en comando, y el resto serán los argumentos.
Una vez que poseemos estas dos variables, estamos en condiciones de ejecutar el comando.
Para encontrar el comando dentro del arreglo, utilizaremos una de las funciones que forman parte de la biblioteca estándar de C, **bsearch**. Esta función necesita de otra que le diga si encontró o no el elemento buscado; en nuestro servidor es **compararComandos** (remitirse a la página del manual para ver cómo funciona bsearch). Si bsearch tuvo éxito, nos devolverá un apuntador al elemento; en caso contrario, devolverá **NULL** (por lo tanto, le enviaremos el error al cliente).
Si encontramos el comando, entonces ejecutaremos la función mediante el apuntador que tiene registrado. De esta forma, podemos agregar comandos a nuestro arreglo de comandos y no tendremos que tocar una sola línea de código para que los encuentre.
Como vimos al principio, al inicializar el arreglo comandos, se pusieron todas las funciones que ejecutan cada comando. Por ejemplo, la función comandoMensaje repetirá el argumento pasado por parámetro a todos los clientes que estén conectados a nuestro servidor.

LECTURA CON BUFFER

Otro de los puntos interesantes es la función cpLeerSocketBuffer, que nos permite hacer el clásico read, pero con un buffer de por medio. Veamos parte del código (archivo **socket_buffer.c**):

```
char *cpLeerSocketBuffer
(SocketBuffer
*stSocket, char *crResultado) {
        unsigned int uiBytes;
        uiBytes = read(stSocket->socket,
stSocket->buffer + stSocket->bytesLeidos,
stSocket->tamanioBuffer - stSocket->bytesL
eidos);

        if(uiBytes == 0) {
                *crResultado = LEER
_SOCKETCERRADO;
                return NULL;
        }
        stSocket->bytesLeidos += uiBytes;
        return cpObtenerComando(stSocket,
crResultado);
}
```

Esta función hace uso de read, que nos permite leer desde un archivo (en este caso serán sockets). Read necesita tres argumentos: el archivo desde donde leer, un buffer en donde escribir lo que leyó, y un tamaño máximo de bytes por leer. Nosotros utilizaremos, como buffer, el buffer (valga la redundancia) desde la última posición ocupada, y leeremos como máximo la cantidad de bytes restantes en él.
Si read devuelve 0, significa que la conexión se ha cerrado por parte del cliente (el if pondrá en el resultado **LEER_SOCKETCERRADO** y devolverá NULL como mencionamos antes). En caso contrario, read, devolverá la cantidad de bytes leídos, que no necesariamente concordará con el valor pedido.
Luego, una vez que el buffer esté algo "completo", lo pasaremos por la función cpObtenerComando, que se encargará de revisar si la línea está completa o no, y devolverá el valor correspondiente.

ENTRADA PARA LA CLIENTELA

Como hemos dicho al principio, sólo hace falta un cliente de Telnet (que viene con cualquier distribución de GNU/Linux). Para ingresar en nuestro servidor (una vez que lo hayamos levantado) tenemos que ejecutar desde la consola:

```
$ telnet 2222
```

, donde 2222 es el puerto en el que lo levantamos (puede ser cualquier número, pero si no se es root, deberá ser mayor que 1024).

EN CONCLUSION

En esta oportunidad hemos analizado un programa con cierto grado de complejidad y hemos combinado diversas herramientas. Usamos select para multiplexar nuestro servidor, atrapamos señales para controlar el fin de la ejecución, y recurrimos a apuntadores a funciones con el fin de crear un interesante sistema que nos permita identificar y ejecutar los distintos comandos, y de que sea sencillo y fácil de expandir. De este ejemplo podrán tomar ideas que les permitan mejorar sus propias aplicaciones, además de que podrán tomar este servidor como base para desarrollar los suyos propios. ★ **Demian Alonso**

SERVIDOR DE E-MAIL - PARTE 2

Courier MTA

EN LA PRIMERA PARTE DE ESTA NOTA LLEGAMOS A CONOCER LOS PARAMETROS MINIMOS NECESARIOS PARA DEJAR A COURIER FUNCIONANDO; SOLO NOS FALTABA CONFIGURAR EL BACK-END DE AUTENTIFICACION. EN ESTA OPORTUNIDAD, VAMOS A TRABAJAR CON LA BASE DE DATOS RELACIONAL POSTGRESQL, UNA VERDADERA MARAVILLA, PARA ALMACENAR LA DEFINICION DE CUENTAS DE CORREO.

Courier puede autentificar usuarios definidos en distintos lugares simultáneamente, como **/etc/passwd**, **/etc/shadow**, o directamente vía PAM o el archivo indexable userdb, mediante vpopmail o LDAP. Sin embargo, en nuestro caso almacenaremos todos los usuarios en una base de datos SQL. Podemos elegir entre MySQL y PostgreSQL, pero de MySQL se ha hablado mucho, y Postgre tiene mucho por ofrecer. Conozcamos algunas de sus características principales antes de continuar.

CARACTERISTICAS DE POSTGRESQL

En este mismo momento (digo, cuando estén leyendo este artículo, claro...) PostgreSQL 8.0 ya habrá sido liberado como versión estable. Veamos qué tiene para ofrecernos (algunos ítems son propios de la jerga de los DBA, y debido a cuestiones de espacio, no nos detendremos a explicarlos):

- Soporte casi completo del estándar SQL 2003.
- Consultas complejas (selects anidados, inner join, left outer join, etc.).
- Foreign keys.
- Triggers.
- Views.
- Integridad transaccional.
- Control de concurrencia multiversión.
- Herencia.

Entre las novedades de la versión 8, podemos mencionar:

- Recuperación por marca temporal.
- Savepoints.
- Tablespaces.
- Funciona nativamente en MS Windows NT y superiores.
- Excelente performance en sistemas de múltiples procesadores.
- Permite cambiar el tipo de dato de una columna.
- COPY acepta archivos CSV.
- Soporte Perl del lado del servidor.
- Mejor soporte JDBC.
- Mejor performance en la administración y el uso de la memoria.

Como lo declara la documentación, uno de los aspectos más interesantes de PostgreSQL es la posibilidad de extenderlo, agregando nuevos tipos de datos, funciones, operadores, métodos de indexado y lenguajes procedurales.
Por supuesto, ustedes pueden preguntarse por qué usar PostgreSQL si es tan poderoso. Dependiendo del servidor o grupo de servidores que estén diseñando, el backend SQL puede ser usado por múltiples servicios, y, si tienen un DBA o piden ayuda al respecto, pueden llevar al mínimo la redundancia de datos.

REQUERIMIENTOS DE CONFIGURACION

Para que Courier-MTA quede funcionando, tenemos que realizar los siguientes pasos:

0. ¡Instalar PostgreSQL! (cuac!).
1. [OPCIONAL] Indicarle a PostgreSQL que acepte conexiones TCP/IP, si va a funcionar en un servidor separado de Courier-MTA.
2. Agregar un usuario de PostgreSQL (¡atención! NO de sistema) para que Courier utilice con el fin de conectarse a la base de datos.
3. Crear la base de datos y la tabla de PostgreSQL necesarias para almacenar los detalles de configuración de cada usuario.
4. Configurar detalles de authpgsqlrc.

Veamos cómo realizar cada uno de estos pasos.

ATENCION: Voy a analizar algunos detalles de configuración considerando una distribución Gentoo. Una forma rápida de instalar Gentoo es instalando UTUTO-e o "s", luego hacer "emerge sync" e instalar los paquetes necesarios: **https://e.ututo.org.ar** o **http://ftp.gnu.org/gnu+linux-distros/ututo-e**.

0. Instalación de PostgreSQL

Si utilizan Gentoo / UTUTO-e o una distribución basada en RPM o Debian, este paso es sencillo. En el caso de RedHat o Fedora, pueden instalar los rpm mediante yum, o siguiendo el método tradicional. En Gentoo, "emerge postgres" es suficiente para instalarlo, pero este paquete necesita ejecutar un comando adicional, para crear ciertos directorios necesarios, entre otras cosas:

```
ebuild /usr/portage/dev-db/postgresql/postgresql
7.4.6.ebuild config
```

¡Reemplacen 7.4.6 por la versión que hayan instalado!

En Debian, el famoso **apt-get** (disponible incluso para SuSE... **apt4rpm**) hará el trabajo. No olviden utilizar chkconfig o rc-update add postgresql default, en el caso de Gentoo, para ejecutar PGSQL al inicio del sistema.

1. (Opcional) Habilitar TCP/IP para PostgreSQL

En Linux hay dos clases de sockets: UNIX y TCP/IP. Los sockets UNIX son archivos físicos creados por un programa que espera conexiones, para que otros programas lean y escriban a él con el objetivo de comunicarse. Los sockets TCP/IP, en cambio, escuchan en una combinación dirección-/puerto. En el caso de Courier, se pueden utilizar ambos mecanismos, pero sólo se podrá usar un Unix Socket si tanto PostgreSQL como Courier-MTA pueden ver un mismo sistema de archivos. Si podemos utilizar esta funcionalidad, y ningún otro programa requiere acceder vía TCP/IP a PostgreSQL, usémosla. En caso contrario, podemos habilitarla mediante el parámetro "-i" del binario postmaster de PostgreSQL.

En el caso de Gentoo, esto es de lo más simple: sólo hay que agregar el parámetro "-i" a la variable PGOPTS, al final del archivo **/etc/conf.d/postgresql**. En el caso de alguna otra distribución, en **/etc/sysconfig** seguramente encontraremos algún archivo denominado pgsql, postgres o postgresql.

2. Crear un usuario de PostgreSQL

Esta parte es verdaderamente sencilla. Una vez iniciado PostgreSQL (**/etc/init.d/postgresql start, rcpostgresql start, service postgresql start**, o lo que sea necesario), debemos hacer **su** al usuario postgres. Desde allí, creamos el usuario "courier" con el comando createuser, de la siguiente manera:

```
murray ~ # su - postgres
postgres@murray ~ $ createuser -ADE -P courier
Enter password for new user:
Enter it again:
CREATE USER
```

La clave del usuario se asigna en forma interactiva, y con los parámetros -A, D y E no le damos ningún privilegio al usuario, y logramos que la clave se almacene encriptada. Este usuario NO aparecerá en /etc/passwd, ya que NO ES de sistema, sino propio de PostgreSQL.

3. Crear base de datos y tablas requeridas

Cuando no existen usuarios de base de datos con privilegios administrativos, el usuario DE SISTEMA bajo el contexto del cual funciona PostgreSQL tiene dichos privilegios. Podremos crear la base desde el contexto de dicho usuario. A tal efecto, usaremos la siguiente combinación de comandos:

```
postgres@murray ~ $ createdb -O courier maildb
"Base de datos para CourierMTA"
```

El comando createdb –help nos mostrará un listado de parámetros útiles. En este caso, usamos –O, que nos permite especificar qué usuario de PostgreSQL será el dueño de la base (en este ejemplo, el usuario courier). La base se llamará maildb y su descripción, que es opcional, será: "Base de datos para CourierMTA". Conviene utilizar **wget** y bajar el archivo con los comandos SQL necesarios para crear la tabla. Puede encontrarlo en: postgres@murray~$ wget www.buanzo.com.ar/files/courier.sql

El archivo courier.sql quedará en el directorio home del usuario postgres.

Una vez creada la base, vamos a conectarnos a ella. Aprovecharemos que ya hemos creado al usuario courier, y que es dueño de la base, para crear la tabla necesaria para el funcionamiento de Courier-MTA:

```
postgres@murray ~ $ psql -U courier -W maildb
Password:
Welcome to psql 7.4.6, the PostgreSQL interactive
terminal.
Type: \copyright for distribution terms
 \h for help with SQL commands
 \? for help on internal slash commands
 \g or terminate with semicolon to execute query
 \q to quit
maildb=>
```

Ahora, desde el prompt del psql podemos ingresar comandos SQL. Utilizamos el siguiente comando para crear la tabla automáticamente:

```
maildb=> \i courier.sql
psql:/tmp/courier.sql:13: NOTICE: CREATE TABLE /
PRIMARY KEY will create implicit index
"id" for table
"cuentas"
CREATE TABLE
```

Con "\d" podemos ver la tabla, llamada "cuentas"; y con "\d cuentas" tenemos acceso a su descripción. Ahora sólo debemos configurar Courier... y tal vez crear una cuenta de mail.

4. Configuración de /etc/courier/authpgsqlrc

De este archivo, primero configuraremos los parámetros vinculados a la conexión con la base de datos. El puerto predefinido de PostgreSQL es el 5432. El archivo ejemplo de courier indica 5400, de modo que debemos corregirlo. Supongamos que vamos a utilizar TCP/IP, pero en el mismo servidor:

```
PGSQL_HOST          localhost
PGSQL_PORT          5432
PGSQL_USERNAME      courier
PGSQL_PASSWORD      clave_asignada
_via_createuser
```

Por supuesto, debemos poner la clave correcta. Recuerden, con el comando psql pueden verificar todos los parámetros de conectividad: puerto, host, usuario y clave. Ahora, seteamos la variable **PGSQL_DATABASE** en "maildb" y **PGSQL_USER_TABLE** en "cuentas". Luego, debemos decidir si las claves se almacenarán encriptadas o no, comentando o descomentando. A nivel seguridad, es mejor encriptarlas. También podemos habilitar los dos campos, pero sólo uno de ellos por usuario podrá tener información. La única ventaja de no encriptar es que podremos autentificar utilizando **CRAM-MD5**. Si no cambiamos el nombre de los campos de courier.sql, los valores predefinidos para todas las variables ***_FIELD** estarán bien. Al configurar un cliente POP3 o IMAP, el nombre de usuario será lo que diga el campo "id": la dirección de mail completa. En **DEFAULT_DOMAIN** podemos setear el nombre de dominio que vamos a agregar si alguien sólo utiliza de usuario el texto a la izquierda de la arroba. Si mantenemos comentada la variable **PGSQL_MAILDIR_FIELD**, el valor predefinido será el de la columna home + /Maildir. Creamos un usuario, que tendrá su mail en **/dominios/buanzo.com.ar/buanzo/Maildir**. El UID y el GID de sistema 65534 deberán tener acceso con una política que le permita al UID leer, escribir y acceder al HOME y al MAILDIR. Lo mismo vale para el GID, pero sin escritura. A la cuenta buanzo@ se le asignan 10 MB de quota (mediante la fórmula MEGAS*1024*1024 y la letra S al final) y una clave 'clave'. El comando SQL por utilizar desde psql será el siguiente:

```
INSERT INTO cuentas(id,clear,home,quota) VALUES
('buanzo@murray.
com.ar','clave','/dominios/murray.com.ar/
buanzo','1048576S');
```

Por supuesto... ¡hay que verificar que **murray.com.ar** o el dominio que utilicemos, sea el que exista en **/etc/courier/hosteddomains**!

¡Nos leemos la próxima! No duden en enviarme sus consultas, porque éste es un tema que da para mucho más. ★ **Arturo "Buanzo" Busleiman**

UN MOMENTO DE REFLEXION SOBRE LA ACTUALIDAD DEL MUNDO DEL SOFTWARE...

Convivencia en paz

LAS VECES EN QUE TUVE LA SUERTE DE MANTENER UNA CHARLA PERSONAL CON RICHARD STALLMAN (PRESIDENTE DE LA FREE SOFTWARE FOUNDATION Y PADRE DE GNU/LINUX), DISCUTIMOS (AMIGABLEMENTE) SOBRE UN MISMO TEMA: EL SOFTWARE PROPIETARIO: ¿DEBIERA NO EXISTIR?

★ "El software propietario es malo y no debiera existir", son las palabras del mismísimo Stallman. El, y muchos otros "evangelizadores" del Software Libre, están totalmente convencidos de que el soft propietario es malo, porque impide el correcto desarrollo de la industria y evita que la información llegue a todos los usuarios, en igualdad de condiciones. Ahora bien, en este afán de "libertad"… ¿no nos estaremos olvidando de la otra parte? Mi planteo es el siguiente: todo programador tiene la libertad de desarrollar la solución de software que quiera y del modo que quiera. Esto es, yo, como programador en lenguaje C, puedo decidir hacer un reproductor de MP3 en donde el botón de Play, en realidad, ejecute una pausa, y el de Stop comience la reproducción. Es una estupidez, claro está, pero nadie puede decirme cómo hacer mi programa, ni qué herramientas puedo utilizar, siempre y cuando esas herramientas sean de libre uso, por supuesto. Ahora, cuando finalizo el desarrollo de mi programa, yo, como único creador, opto por definir los medios de distribución. Puedo ofrecerlo "en mano", distribuirlo por Internet, venderlo en cajitas de colores... puedo hacer lo que quiera, por la simple razón de que yo soy su creador y, si registré legalmente el código, soy su dueño ante la justicia. Entonces, cuando opto por un medio de distribución, puedo optar por el Software Libre, o puedo optar por un medio que no lo sea… ¡soy libre! ¿Es malo que exista el software propietario? ¡No, lo malo sería que no existiera la posibilidad de otra cosa que no fuera el Software Libre!

Cuidado, no me malinterpreten. Soy un "evangelizador" del Software Libre como muchos de ustedes, amigos lectores, pero considero que la verdadera libertad es aquella que nos permite definir cualquier regla de juego, sea buena o mala (ése será tema de otra discusión). Si un programador opta por distribuir su creación bajo una licencia propietaria, es su problema (algún día se dará cuenta de que hubiera sido mejor hacerlo vía Software Libre), ¡pero aplaudo la libertad que tuvo de tomar esa decisión!

SOFTWARE PRIVATIVO EN GNU/LINUX

★ WINE, DosEmu, CEDEGA, VMWare, Bochs y tantos otros son emuladores que nos permiten correr en GNU/Linux aplicaciones diseñadas para otros sistemas operativos. Muchos dicen que esas aplicaciones atentan contra la causa del Soft Libre, ya que permiten ejecutar software privativo en GNU/Linux, y eso es malo. ¿Malo? ¿Malo es que un diseñador que migró a GNU/Linux pueda utilizar su herramienta de diseño (como Dreamweaver) privativa, pero que pagó, en su sistema libre? ¡Malo sería si no la hubiera pagado! ¡Malo es que no haya una opción de igual calidad en el mundo del Software Libre! (seguramente aparecerá en poco tiempo más). Yo considero que estas herramientas son una solución para aquellos que tienen la intención de usar GNU/Linux, pero que, por cualquier razón, no pueden dejar de utilizar la aplicación que usan actualmente.

SOFTWARE LIBRE EN WINDOWS

★ "Aplicaciones de Software Libre en el escritorio de Windows representan una situación en la que el Open Source no puede salir vencedor", dijo **Aaron J. Seigo**, un desarrollador del proyecto KDE. Su frase viene a colación de que hay muchos proyectos de KDE que están siendo "portados" a Windows. En parte, es verdad que los usuarios actuales tienen pocas razones para dejar de usar su actual Windows. Pero, mi opinión, "mostrarles" un poco de Software Libre en su sistema privativo puede ser una buena forma de que se pregunten: "ah bueno, si esto es así, ¿como será el paquete completo?". O sea, es una forma de iniciar en este fascinante mundo a gente que no tiene absolutamente nada que ver con el Software Libre.

MI EXPERIENCIA PERSONAL

★ Luego de varios años de trabajar con herramientas de Software Libre me doy cuenta de que los usuarios que no tienen interés nato en él, comienzan a tenerlo cuando conocen algunas de esas herramientas. Por ejemplo, cuando ven GIMP funcionando en Windows, ya tienen ganas de instalar GNU/Linux para usar una versión mucho más optimizada. Por otra parte, conozco muchas PyMEs que utilizan software privativo especialmente diseñado para sus necesidades, y que quizá funciona en sólo DOS y Windows. Si no pueden dejar de utilizarlo por ser su herramienta principal de trabajo, entonces los emuladores como Wine (para Windows) y DosEmu (para DOS) son una excelente opción: pueden seguir utilizando la herramienta de siempre en un sistema operativo mucho más estable del que venían usando.

¡EQUILIBRIO!

★ Mi opinión (y espero que también las de ustedes, amigos lectores) es que todo debe llevarse con su debido equilibrio. ¿El software privativo es malo? No, no creo que le haya pegado a nadie… Pero, prefiero la libertad del Software Libre toda la vida. Lamentablemente, hay casos en los que el Software Libre no puede ofrecernos una solución, y entonces debemos recurrir a herramientas de emulación o utilizar por lo menos algunos componentes libres sobre un sistema propietario. En fin, el software privativo nunca dejará de existir, y mucho menos el libre. Entonces, mientras la opción privativa se presente como ventajosa para algunos programadores… ¡que ambas convivan en equilibrio!

★ **Facundo Arena.**